Metisse » XHTML 2 » Greenphone » Mono

LXF DVD

Тройная загрузка!
» Mandriva One Metisse
» OpenSUSE 10.2 Live
» GNewSense 1.0

LINUX FORMAT

Главное в мире Linux

Системы управления версиями: Subversion и CVS – лицом к лицу с. 20

Апрель 2007 № 4 (91)

План занятий на апрель

Разбираемся с файлами журналов с. 38
Создаем таблицу стилей XSLT с. 60
Настраиваем SugarCRM с. 64
Собираем Deb-пакеты с. 72

Открывая Gimp заново

Будущее свободной графики – в ваших руках!

» Новые инструменты Gimp 2.4 » Вдохновляющие проекты других пользователей » GEGL из первых рук

Webатастика!

Nokia N800 на базе Linux – Интернет в кармане брюк
с. 12

Linux center
www.linuxcenter.ru

“Я удивлен – существует не так много технологий, насчитывающих четверть века, которые до сих пор работают”

Эрик Оллмен о Sendmail с. 28

4 607130 820506

LINUX FORMAT

Апрель 2007 № 4 (91)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

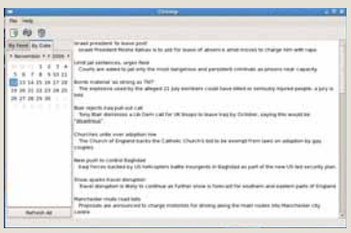
Учебники

Система
Следите за пользователями! 56
Стойкие пароли и права доступа к файлам для начинающих.

XSLT
Создаем таблицу стилей 60
Программирование на языке таблиц стилей XML: ваша RSS-лента будет выглядеть намного лучше.

SugarCRM
Как ее настроить 64
Мы установили великолепную открытую CRM-систему: пришла пора приспособить ее к нуждам вашей организации.

Mono
Программируйте с GTK 68
Набросайте интерфейс для своего RSS-агрегатора и заставьте его работать!



› **Сделайте это в Mono!**

Deb
Соберите собственный пакет 72
«Создание и распространение пакета – это способ выразить благодарность сообществу». Узнайте больше!

GTK
Новый Glade 76
Сделайте это на C! Новый Glade позволяет писать удивительно изящные приложения.

Unix API
Окна ncurses 80
Создаем удобный интерфейс для утилит командной строки.

Java EE
Сессии и фильтры 84
Разграничиваем доступ к web-приложению и фильтруем нежелательных посетителей.

PostgreSQL
Дополнительные главы 88
Резервное копирование, репликация, автовакуум – что еще умеет PostgreSQL?

LaTeX
Создаем презентацию 94
Чтобы впечатлить аудиторию, не нужно открывать Impress. Сделайте презентацию в LaTeX – и вас не забудут!

Blender
Скелетная анимация 98
Орлята учатся летать – пингвины учатся ходить.

LXF DVD91

Майк вам покажет 116

Mandriva One Metisse

Специально для истинных гурманов: восхитительная «живая» Mandriva с новым трехмерным рабочим столом, Metisse.

OpenSUSE 10.2 Live

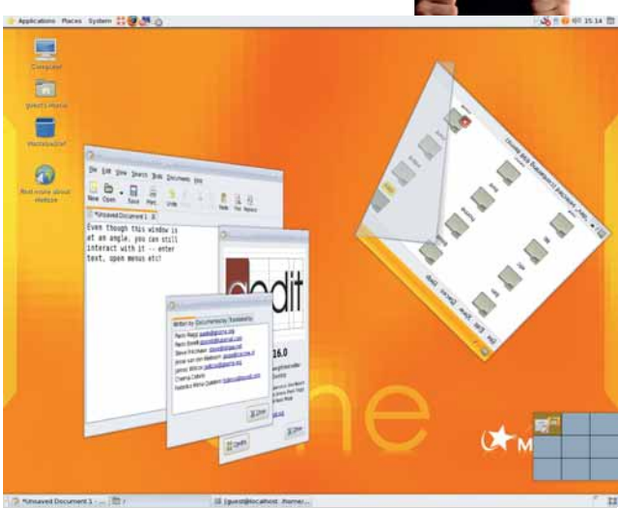
Один из самых солидных дистрибутивов Linux, готовый к запуску прямо с DVD! Покажите Linux своим друзьям, используйте его как спасительный диск или «пробник».

GNewSense

Устали от угрызений совести из-за стоящего на вашем компьютере проприетарного ПО? GNewSense, новый дистрибутив Free Software Foundation, подарит вам долгожданный душевный покой.

Linuxpedia, Ответы и PDF

Узнайте что-то новое и решите ваши проблемы с Linux.



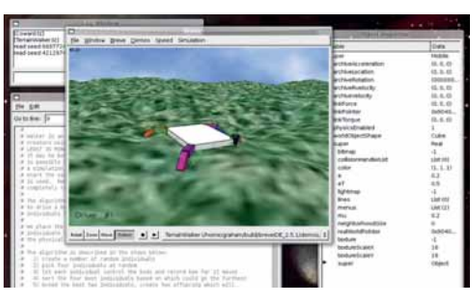
› **Попробуйте новую Mandriva – теперь окна можно гнуть!**

Что за штука... XHTML 2?

Забудьте свои дурные HTML-привычки с. 32

LXFHotPicks

Лучшие новинки открытого ПО на планете 110

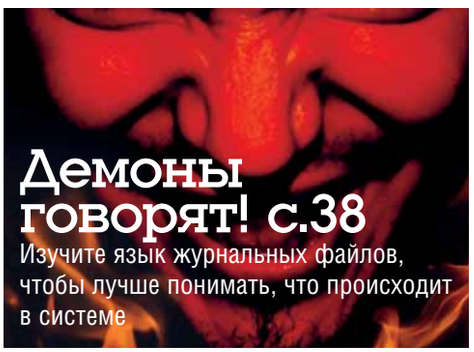


› **Наслаждайтесь искуственным интеллектом с** *Breve.*

LXF ИНТЕРВЬЮ

«Существует не так много технологий, насчитывающих четверть века, которые до сих пор работают.»

Эрик Оллмен с. 28



Демоны говорят! с.38

Изучите язык журнальных файлов, чтобы лучше понимать, что происходит в системе

Подпишись на Linux Format и сэкономь!

Contents

LXF DVD внутри!

См. страницу 116

Спецрепортаж

ВТОРОЕ пришествие GIMP

Что нам готовит фаворит открытой графики? с.48

А также...

15 мгновений безопасности ..с 34
Держите систему на замке

Дневники демонов .....с 38
Ведение журналов: syslog и окрестности

Neo1973 – новая мобильная революция.....с 42
Узнайте, как OpenMoko и FIC изменят вашу мобильную жизнь к лучшему

Путь к просвещению.....с 44
Под знаменами E17 на PS3

Постоянные рубрики

Новости.....04

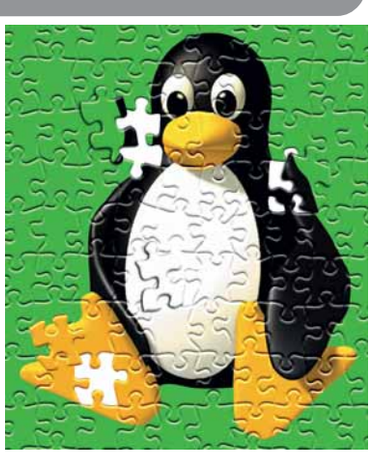
DistroWatch.....26
Дополнительная информация о трехмерном рабочем столе Mandriva плюс SimplyMepis и FreeBSD.

Интервью LXF.....28
Что изменилось бы в Sendmail, если бы Эрик Оллмен начал с нуля?

Что за штука.....32
W3C норовит пройтись метлой по пыльным уголкам Web: что нам готовит XHML?

Ответы.....103
В этом месяце: символьные ссылки, слайд-шоу, Evolution и другие проблемы пользователей

Через месяц.....128
Что ждет нас в мае



Обзоры

Nokia N800.....12
N770 понравился многим – какое впечатление произведет на нашего обозревателя его последователь?

› **Шаловливые ручки Linux Format дорвались и до благопристойного Greenphone.**

HP DVD940e.....14
HP выпустила библиотеку для поддержки технологии LightScribe – мы оценили результаты

K3b 1.0.....15
Утилита записи дисков из KDE доросла до версии 1.0 – насколько номер соотвествует ее зрелости?

Greenphone.....16
Открытый сотовый телефон или игрушка для разработчиков? Мнение Linux Format.

Metisse.....17
Узнайте, чем решение Mandriva отличается от Xgl и AIGLX.

Sabayon Linux 3.2.....18
Linux на десерт – что получится, если смешать все вкусные новинки в одном дистрибутиве?

Сравнение: системы управления версиями

RCS.....21
CVS.....21
Aegis.....22
Monotone.....22
Subversion.....23
Git.....24
Bazaar.....24

ГЛАВНЫЕ НОВОСТИ: » Red Hat Enterprise Linux 5 и Mandriva Corporate Desktop 4.0 Beta » Dell преподносит сюрпризы » Солидный патронаж » Кто сделает Debian привлекательнее?

» Рубрику ведет Илья Шпаньков

Солидный патронаж

Прошедший месяц принес сразу две новости, которые по сути можно отнести к однотипным, хоть и разным по целям и значимости. Поэтому позволим себе объединить их в одну заметку и начнем с наименее претенциозной. Итак, норвежская компания Trolltech, давно и плодотворно сотрудничающая с разработчиками KDE, решила еще больше сблизиться с авторами одной из самых популярных рабочих сред и объявила о том, что становится официальным спонсором проекта. Следует пояснить, что, несмотря на довольно прочный фундамент, данный проект является некоммерческим и любая помощь со стороны, финансовая или техническая, может только приветствоваться, о чем говорит даже специально для этих целей разработанная KDE система патронажа, позволяющая привлекать сторонних спонсоров к более тесному сотрудничеству. Впрочем, до настоящего момента воспользоваться этой возможностью и стать «патроном KDE» пожелал только известный миллионер и вдохновитель другого свободного проекта Ubuntu Марк Шаттлворт. Сегодня он – официальный частный спонсор и, к сожалению, пока единственный.

Таким образом, компания Trolltech стала первым корпоративным патроном KDE и, пожалуй, самым значимым. Именно программисты этой небольшой норвежской фирмы являются создателями графических библиотек и средств разработки Qt, на которых, собственно, и построен весь графический интерфейс KDE. Получение же официального статуса «Patron of KDE» является взаимовыгодным для обеих сторон: свободный проект получает мощную техническую и финансовую поддержку, которые позволят повысить качество конечных продуктов и ускорить разработку в целом, а коммерческая компания взамен получает дополнительные «производственные силы» в лице многочисленного сообщества программистов и обычных пользователей, работающих по принципам Free Software. Это будет способствовать ускоренному внедрению в программные продукты Trolltech новых функций, их тестированию и исправлению ошибок.

Второй случай патронажа коммерческой компании над свободным проектом является не только событием более крупным, но и преследует более замысловатые цели. В двух словах: компания Sun Microsystems стала патроном Фонда Свободного ПО – Free Software Foundation (FSF). Впрочем – не единственным: среди корпоративных патронов Фонда уже числятся такие «мэтры», как Google, Nokia, IBM, Cisco и Intel. Подобный патронаж также осуществляется на взаимовыгодных условиях: компании предоставляют Фонду финансовую помощь, а взамен получают всесторонние услуги консультационного плана, связанные с возможными юридическими и правовыми проблемами, могущими возникнуть в процессе работы со свободным ПО.

В общем, ничего особо сложного и загадочного в данном патронаже нет, но именно случай Sun следует рассматривать более внимательно, учитывая тактику и стратегию действий компании в последние несколько лет. Свой разворот в сторону свободного ПО Sun начала еще в 2004 году (если не считать многолетней поддержки проекта OpenOffice.org) и за прошедший период многие популярные программные продукты компании были выпущены под свободной лицензией. Но изюминка в том, что таким образом известный разработчик ПО не старается придать развитию Linux-систем дополнительный импульса, а решает, грубо говоря, свои проблемы, связанные с недостаточно активным распространением собственной операционной системы Solaris. Причем, одним из возможных сценариев будущего развития компании может стать создание операционной системы GNU на ядре Solaris, что некоторые расценивают как камень в огород «сердца» практически всех свободных систем – ядра Linux. В этом ракурсе патронаж над FSF может оказаться попыткой получить большее влияние на все движение Free Software в целом, а уж как это влияние будет использоваться руководством компании – покажет время.

RHEL 5 – ответный ход

Сильный игрок никогда не будет раскрывать карты раньше срока, а выложит козыри в тот момент, когда соперник уже будет мысленно праздновать победу. Рынок – это тоже игра и мартовский релиз долгожданной операционной системы Red Hat Enterprise Linux 5.0 сопровождался таким солидным набором «козырей», что все предшествующие шаги конкурентов, после которых некоторые аналитики поспешили объявить о скором закате Red Hat, стали выглядеть как-то уж совсем безобидно. Итак, что же приготовил на этот раз лидер рынка коммерческих Linux-систем?

Евгений Максимов (IBM, в центре) рассказывает о сертификации Red Hat. ***Фото Александра Бокового***

Не углубляясь в технологические тонкости, можно отметить четыре наиболее заметных новшества дистрибутива. Прежде всего – это изменение ценовой политики в сторону повышения гибкости. Теперь потенциальный покупатель может легко варьировать свои затраты на приобретение и поддержку операционной системы в зависимости от собственных нужд. Другими словами – он покупает только то, что ему действительно нужно для полноценного ведения бизнеса, не переплачивая за неиспользуемые компоненты или сервисные услуги. При желании их всегда можно будет докупить позднее. Второй выигрышный момент тесно связан с первым: клиенты получили возможность конфигурировать свои IT-инфраструктуры динамически, постепенно наращивая функциональные возможности существующих сетей в зависимости от текущих потребностей. Эти два качества делают новый продукт Red Hat более привлекательным как для крупного бизнеса, так и для молодых компаний, только начинающих свой путь к завоеванию рынка.

Впрочем, более важным качеством, чем цена и масштабируемость, является способность полноценного взаимодействия с программными продуктами других компаний. И в этом плане новой версии RHEL есть чем похвастаться. В процессе подготовки релиза создатели Red Hat плотно взаимодействовали с другими разработчиками, в результате чего операционная система получила ряд значительных дополнений, позволяющих органично внедрять Linux-решения в традиционно «агрессивной» окружающей среде. И в первую очередь это касается полноценной поддержки виртуализации Xen, над которой разработчики RHEL трудились совместно с компанией IBM (последняя, кстати, заканчивает тестирование новой системы на своих аппаратных решениях и обещает полноценную поддержку). Можно отметить, что включение в дистрибутив хорошо развитого средства виртуализации позволило убедить многих сторонних производителей популярного корпоративного ПО обеспечить интеграцию своих продуктов с RHEL. Например, компания Symantec совместно с Red Hat обеспечила совместимость своих Veritas Storage Foundation, Veritas Cluster Server и Veritas NetBackup с новой операционной системой и планирует осуществлять техническую поддержку клиентов, использующих в работе оба программных продукта.

И, наконец, еще одно важное новшество, появившееся с выходом RHEL 5, позволяет предложить потенциальным клиентам компании не просто операционную систему, а целый пакет разработок, позволяющий практически на 100% обеспечить все потребности бизнеса. В рамках проекта, получившего название Red Hat Exchange (RHX), компания будет обеспечивать пользователей не только собственными программными продуктами, но и разработками других компаний, работающих по принципам FOSS и ориентированных на корпоративного клиента. В частности, среди наиболее известных названий можно упомянуть MySQL, Enterprise DB, JasperSoft, SugarCRM, Zenoss, Zimbra, Zmanda, Compiere, Groundwork Open Source и Centric CRM. При этом Red Hat гарантирует полноценное функционирование данных решений в среде RHEL и обеспечивает весь пакет обслуживания. Другими словами, разработчики подготовили не столько новую операционную систему, сколько целый набор мощных и хорошо развитых программных средств, способных обеспечить самые разнообразные потребности IT-инфраструктур бизнеса «из коробки».

www.redhat.com/rhel

Пока верстался номер

Компании IBM и Red Hat объявили о получении государственного сертификата ФСТЭК операционной системы Red Hat Enterprise Linux на серверах IBM System x, IBM BladeCenter, IBM System p, IBM System i и IBM System z. Операционная система Red Hat Enterprise Linux AS, WS Version 4 Update 1 + Audit Pack была сертифицирована на всех серверных платформах IBM и имеет оценочный уровень доверия 4 (ОУД) (усиленный компонентом систематического устранения ошибок) и уровень контроля отсутствия недекларированных возможностей 4 (НДВ) согласно сертификату соответствия ФСТЭК (Гостехкомиссии) России. Получение сертификации данного уровня дает возможность использовать все линейки серверов IBM System с ОС Red Hat Enterprise Linux для работы с конфиденциальной информацией и создания автоматизированных систем.

Процесс сертификации по требованиям ФСТЭК включает в себя детальное исследование соответствия предоставленного исходного кода, а также серьезный анализ документации, описывающей все аспекты разработки, тестирования и поддержки дистрибутива, выпуска обновлений. Сертификационные испытания проводились «Центром безопасности информации» (ЦБИ), аккредитованной организацией на проведение подобных работ ФСТЭК.

Mandriva Corporate Desktop 4.0

В борьбу за корпоративного клиента собирается включиться еще один разработчик популярных Linux-дистрибутивов – компания Mandriva. Об этом недвусмысленно говорит выпуск первой бета-версии Mandriva Corporate Desktop 4.0. Для большего соответствия запросам корпоративного пользователя данный дистрибутив содержит несколько новых компонентов, включающих средства интеграции с мобильными сетями третьего поколения 3G, интерактивные средства настройки сетевой безопасности системы и подключения к виртуальным частным сетям (DrakVPN), средства аутентификации пользователей на основе OpenLDAP или Active Directory и многое другое.

О серьезности намерений говорит также и то, что при подготовке данной версии авторы активно сотрудничали с такими компаниями, как Intel, HP, NVIDIA, Arkeia, BitDefender и VMware. Основная цель такого взаимодействия – достижение максимального соответствия создаваемого ПО аппаратной начинке современных компьютеров и улучшение взаимодействия с другим ПО с использованием средств виртуализации. Дополнительно руководство Mandriva предполагает привлечь к активному тестированию и отладке нового дистрибутива и множество индивидуальных программистов – для этих целей открыта специальная программа Corporate Desktop 4 Beta Program, обещающая участникам множество бонусов и скидок при покупке других продуктов Mandriva.

corpo.mandriva.com/xwiki/bin/view/CD4/

Тот, что осилил дорогу...

LINUX FORMAT Интервью

Пока на Linux-форумах гремят баталии по поводу того, как осчастливить школы свободным ПО, **Константин Владимирович Панфилов** действует. Учитель информатики из Гоноховской средней школы Каменского района Алтайского края уже перевел свой компьютерный класс (что составляет 75% от всего компьютерного парка школы) на Linux. О том, почему он это сделал, и как ему удалось пройти этот путь, и пойдет речь в данном интервью.

Призываю вас не судить реальность с высоты своего опыта — она такая, какая она есть. Мы можем ее только улучшить, а прошлое изменить не получится.

Евгений Балдин: Когда и где впервые услышали/прочитали про GNU/Linux? Почему решили разобраться в нем?

Константин Панфилов: Эта история начиналась где-то два года назад. Одним зимним вечером я сидел в кабинете информатики, и мне в голову постучалась мысль: «А почему-бы тебе не попробовать поставить ОС UNIX?»[1]

На следующий день я обошел всех своих знакомых с вопросом: «Где взять такую вещь?» В ответ пожимали плечами и крутили у виска.

Меня это еще больше раззадорило. Один товарищ сказал: «О Linux слышал, о UNIX – нет». На тот момент я и понятия не имел, что и как – например, какие дистрибутивы существуют. Зайдя на Центральный узел связи, спросил у программиста: «Есть такая вещь?» В ответ я услышал странный и, как мне показалось, ругательный набор слов: «Мандрак нелокализованный на 6 дисках». Ответ меня не устроил, и я подался дальше искать счастья. Повезло мне в банке. Ребята там работают общительные и смекалистые.

Евгений: Так каким был Ваш первый дистрибутив GNU/Linux и почему?

Константин: В банке мне дали ASPLinux 9.0 Chip Edition на одном CD. Дистрибутив оказался урезанным, но эта система меня очень заинтересовала. С этого все и началось.

Евгений: А как GNU/Linux оказался в школе?

Константин: Где-то спустя два месяца после первого знакомства я приобрел Mandrake 10 на трех CD. Дистрибутив был более дружелюбен к пользователю. Тогда в школе стояло 4 машины. Сразу захотелось опробовать сетевые возможности ОС. Была поставлена Samba, запущена сетевая печать. Но знаний было мало. Раздобыл два руководства по администрированию, у кого-то из друзей нашел подборку о Linux на CD – вот так и началась моя дорога в Linux.

Но тогда система еще не использовалась в школе в полную силу. На тот момент я отвечал за обработку документов для администрации, учителей, а поток шел большой. Нужен был аналог Fine Reader'а, а его, как известно, нет до сих пор.

Чуть позже наткнулся на ALUG (Алтайская LUG http://alug.metroland.ru/), LinuxForum (http://linuxforum.ru/). Сначала «потусовался» на ALUGe, но там было скучновато. Больше понравилось на LinuxForum. Там бурлит настоящая жизнь. Практически на любой вопрос можно найти ответ, корифеи всегда помогут и поправят. Здесь я понял: «Linux в школе должен быть». Мне нравится эта система своей открытостью, идеологией.

[1] Читал статью Максима Левина «Как стать хакером».

Евгений: Были ли проблемы при переходе?
Константин: Первая попытка была полтора года назад – но не вышло. Это было обусловлено парком техники и дистрибутивом (Mandrake 10 Discovery). Хотя в локальной сети был поставлен *Apache* с целью научить детей работать в браузере, дать понятие, что такое Сеть. Ведь подключения к Интернету не было, да и сейчас его фактически нет. Линии у нас абы какие – можно сказать, не проложены. Окончательный переход состоялся уже в этом году.

Евгений: Какая причина привела к этой достаточно резкой замене программного обеспечения?
Константин: Резкой заменой я бы это не назвал. В данном случае, все назревало постепенно. Ну и конкретный повод все-таки был: события в стране, в частности в Пермском крае. Да и техника у нас сейчас стоит нормальная. Если еще учесть экономический эффект, то переход можно назвать логическим следствием. Хотите – называйте резкой заменой, ваше право.

Евгений: Вернемся еще раз к вопросу о ПО. Что используется? Какие программы оказались очень кстати?
Константин: Насчет дистрибутива: пробовал ASPLinux, Fedora Core, Mandrake, Mandriva 2006. Больше всего понравился Mandrake/Mandriva своей дружественностью к начинающему пользователю. В качестве рабочего стола – KDE.

Что касается программ: *OpenOffice.org* – альтернатива *Microsoft Office*, это из привычных. Для себя сделал открытие (и всем советую), текстовый процессор *LyX*. С ним действительно избавляешься от болезней *Word*. Очень кстати оказались пакеты *KEduca* с редактором тестов и *KTurtle* – аналог логомиров. Аналог *Delphi* – *Lazarus* плюс *Free Pascal*. Ну и вообще, все что есть в Mandriva, очень кстати.

Евгений: Что заменить не удалось? Чего Вам сейчас не хватает?
Константин: Вот чему не удалось найти замену – так это *FineReader*'у.

Чего пожелать еще? Чтобы Министерство образования развернулось в сторону Open Source, или хотя бы чуть поправило свой курс в эту сторону.

Евгений: Как отреагировали на эти изменения дети?
Константин: Детям интересно. Примерно 80% смотрят заинтересованно, сравнивают и делают для себя выводы. И в пользу этой ОС. Что разочаровывает их, так это ответ на вопрос: «А есть в продаже такие же игры, как под «винду»?»

Евгений: Попробовал ли кто из детей GNU/Linux дома?
Константин: Пока только двое поставили эту ОС. Причина проста – нужны опыт, знания и желание. О чем говорить, если мы только-только перешли на Linux. Но помощь с моей стороны – это святое дело.

Евгений: Как реагируют родители?
Константин: Родителям все равно. Многие из них не имеют высшего образования. К компьютеру относятся, как к дьяволу [улыбается].

Евгений: Как относятся к переходу на GNU/Linux другие учителя?
Константин: Учителя пока «ни сном, ни духом». Да и мало кто из них использует компьютерную технику в учебном процессе. Ведь в массе своей это люди 45–55 лет, которым трудно вводить в свою работу что-то новое.

Когда этот контингент заканчивал учебные заведения, массовой информатизации не было. А чтобы освоить технику сейчас – нужно время и желание, а инертность и боязнь показаться безграмотным берут свое. Но это все временно и преодолимо.

Реально сейчас на технике работают 20% коллектива. Но к концу года эта цифра изменится – я думаю, до 60%. У нас ведь не только стационарные ПК: есть ноутбук и медиапроектор, которые в последний месяц используются весьма интенсивно.

Вот здесь можно снова обратиться к вопросу о программном обеспечении: нужен хотя бы и коммерческий учебный софт, аналогичный существующему под Windows. Зачем Министерству образования тратить деньги на Microsoft? Заплатили бы лучше своим программистам, пусть делают учебное ПО под Linux. Еще и сэкономят.

Евгений: Есть ли помощь от сетевого сообщества?
Константин: В сетевом сообществе уже есть течение в нужном направлении. Помощь – ссылки на ресурсы Интернет. А по части обмена опытом – пока нет.

Евгений: Какие у Вас теперь планы?
Константин: Планы, можно сказать, наполеоновские. Внедрить ОС в учебный процесс в нашей средней школе. Дать заинтересовавшимся ребятам основы PHP, MySQL и администрирования ОС. Учить основам web-дизайна. Это пока все. Дальше мечтать не буду – а то не сбудется [улыбается]. LXF

Сюрпризы DELL

Перестановки в руководстве компании DELL, как и любые другие кадровые перемены, были нацелены на изменение тактики и стратегии в целях улучшения сбыта производимой продукции и, естественно, роста прибыли. Новое начальство решило взять на вооружение схему, давно и успешно отработанную во множестве проектов, разрабатываемых по принципам FOSS – привлечь к участию в дальнейшем развитии компании огромное сообщество пользователей, покупающих компьютеры производства DELL. Само собой, коммерческая направленность деятельности этого всемирно известного изготовителя вычислительной техники не позволяет полноценно вовлекать своих потребителей в производственный процесс, но даже качественная обратная связь может оказать неоценимую помощь в выработке верной стратегии развития. Другими словами, чтобы максимально удовлетворить спрос покупателя, разработчик решил узнать – а что же, собственно, хочет видеть этот самый покупатель в компьютерах и ноутбуках с логотипом DELL? Сказано – сделано, и вот уже в Сети открыт новый ресурс под названием «Dell Idea Storm», на котором все пользователи могут высказать свои пожелания гиганту индустрии. Идея отличная, реализация – на уровне, но никто из руководства DELL не ожидал того сюрприза, что преподнесет новый web-сайт уже через несколько дней после открытия.

Ко всеобщему удивлению, почти две трети всех ответов содержали просьбу о выпуске в продажу компьютеров с предустановленной операционной системой Linux. Одна треть людей также предлагала снабжать новые машины свободным пакетом офисных приложений *OpenOffice.org*. К моменту написания данной статьи уже более 112 тысяч пользователей просят Linux, около 80 тысяч хотят работать с документами в свободном офисе, а более 50 тысяч людей предлагают вообще не устанавливать никакой операционной системы, оставляя за пользователем право выбора. И это при общей цифре, едва превышающей 180 тысяч пожеланий! Конечно, можно предположить, что в силу «врожденной» повышенной социальной активности сторонников свободного ПО, данный опрос нельзя считать стопроцентно отражающим рыночный спрос, ведь Windows-пользователи могли просто промолчать в силу того, что их и так все устраивает. Но факт остается фактом: тысячи людей просят Linux и не принимать этого в расчет нельзя, т.к. в противном случае весь проект под оптимистичным названием «Dell Idea Storm» просто теряет смысл.

Ответная реакция DELL показала, что организаторы опроса не были готовы к подобным результатам. В прессе начали появляться противоречивые высказывания представителей солидной компании, в которых одни заявляли, что компания уже готовится выпускать компьютеры и ноутбуки с предустановленной операционной системой Linux и в данный момент просто пытается выбрать подходящий вариант, другие же советовали не искать в ближайшем будущем на прилавках появления «заряженной пингвинами» техники DELL на прилавках магазинов, связывая это с необходимостью долгого и тщательного анализа соответствия свободного ПО и аппаратной начинки продукции компании. Впрочем, постепенно страсти улеглись и началась спокойная работа по претворению пользовательских идей в жизнь. Именно об этом говорит тот факт, что компания организовала вторую серию опросов (уже на собственном «фирменном» web-сайте), в которых все заинтересованные могут более подробно указать, какой Linux нужен, с какими возможностями и в каком виде. Таким образом, пусть и без указания точных сроков, но предсказать выпуск в продажу компьютеров DELL с предустановленной системой Linux можно вполне определенно.

www.dellideastorm.com

Новости короткой строкой

» Компания Sun Microsystems организовала бесплатную рассылку OpenSolaris Starter Kit на DVD носителях всем желающим.

» Red Hat подписала договор о сотрудничестве с компанией VMware, чтобы обеспечить более качественное взаимодействие популярной виртуальной машины с ОС Red Hat.

» Проект Mono объявил о включении в свою разработку полноценной поддержки популярной среды программирования Visual Basic.NET.

» Правительство республики Куба приняло решение перевести все компьютеры в стране с проприетарной и «идеологически враждебной» Windows на использование свободного ПО на базе GNU/Linux.

» Компания Hewlett-Packard сообщила, что в скором будущем планирует выпустить на рынок ноутбуки с предустановленной операционной системой Linux.

» Компания Linspire переходит на использование Kubuntu в качестве базовой платформы для своей ОС. Это связано с лучшей поддержкой KDE, чем в выбранном ранее на роль базового дистрибутиве Ubuntu.

» Французские парламентарии определились с выбором свободной операционной системы, призванной заменить Windows на 1200 компьютерах народных избранников – вопреки ожиданиям, это будет не отечественная Mandriva, а Ubuntu.

Кто сделает Debian более привлекательным?

Так сложилось, что проект Debian стоит особняком от всех популярных дистрибутивов. И дело не в том, что это один из старейших проектов, использующих свой собственный «именной» формат бинарных пакетов **.deb**. Гораздо больше на обособленность дистрибутива влияет его консерватизм, выраженный в маниакальном соблюдении лицензионной чистоты входящих в состав пакетов и стремлении к максимальной стабильности даже в ущерб желанию пользователей получать наиболее свежий софт. Эти и многие другие, менее заметные качества проекта, послужили тому, что сегодня имя «Debian» ассоциируется с чем-то массивным, неповоротливым, да еще и преклонного возраста. Возможно, на данный неблагоприятный имидж влияет и тот факт, что на сегодняшний день в состав данного дистрибутива входит наибольшее число программ: полный срез репозитория Debian уже давно не укладывается и в дюжину CD-дисков. А ведь нужно еще и отслеживать ошибки в тысячах приложений и своевременно исправлять их. С таким объемом работы делать это довольно сложно, в результате – сроки релизов постоянно откладываются, а ошибки ликвидируются уж совсем черепашьими темпами.

Впрочем, разработчики и сами понимают, что на фоне шустрых собратьев по цеху развиваться таким образом и дальше – непозволительная роскошь. Нужно искать новые решения и одним из таких нововведений стал открывшийся недавно ресурс под названием «Debian Package of the Day». Здесь ежедневно публикуется подробное описание одного из входящих в дистрибутив пакетов, сопровождаемое экранными снимками или наборами управляющих команд. По задумке авторов инициативы, таким образом пользователи смогут избавить себя от мучительного поиска нужных программ среди тысяч входящих в дистрибутив путем установки и тестирования – гораздо проще вначале ознакомиться с описанием, а уж потом решать, стоит устанавливать этот пакет или нет. Правда, и здесь не все так просто: на сайте пока нет даже рубрикатора, позволяющего быстро находить приложения по функциональным признакам, а глобальный поиск в архивах не всегда дает удовлетворительные результаты.

На самом деле, подобные редкие новшества вряд ли помогут кардинально изменить ситуацию. Нужна глобальная программа реорганизации проекта и сейчас – самый подходящий момент сделать это. Каждый год участники проекта выбирают лидера, который будет возглавлять многочисленный коллектив разработчиков и направлять их в определенное русло в соответствии с собственной программой развития. Не является исключением и год текущий: на 18 марта назначено начало периода трехнедельного голосования за одну из представленных кандидатур. В этот раз на пост лидера выдвинуто, пожалуй, наибольшее число претендентов за все время существования проекта – восемь человек (обычно 3-4 кандидата), что еще раз подтверждает сложность текущего положения дел. Каждый кандидат опубликовал на сайте Debian свою программу, в которой отметил наиболее приоритетные направления развития на ближайший год. Несмотря на множественные мелкие различия, практически все кандидаты солидарны в одном: нужно сделать дистрибутив более привлекательным для потенциальных пользователей. А вот как это будет происходить и кто возьмется выполнить сию нелегкую задачу – покажут выборы. debaday.debian.net LXF

Технология счастья

SUNRADIO.RU

сетевое радио под ключ на базе Linux
новое будущее вашей компании

pr@sunradio.ru +7 812 955 76 70 www.sunradio.ru

КОРПОРАТИВНЫЕ СЕРВЕРЫ
И СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ

(812) 327-5960
(495) 232-9230
info@trinitygroup.ru

TRINITY
CORPORATE IT PROJECTS

- Анализ существующей IT инфраструктуры
- Разработка технического задания
- Проектирование, монтаж, внедрение
- Комплексное управление IT инфраструктурой
- Катастрофоустойчивые решения

Серверы

под Linux
FreeBSD
Solaris x86

для баз данных, интернет шлюзов, WEB–приложений, кластеры для научных расчетов

Мы делаем бизнес успешным

www.trinitygroup.ru
Информационные технологии от экспертов

17-19 мая 2007 года, Санкт-Петербург, Россия

Место встречи «инновационных» Востока и Запада

ОТ НАУКИ К БИЗНЕСУ

I Международный Форум
ОТ НАУКИ К БИЗНЕСУ
«Международный опыт развития инфраструктуры инновационной деятельности»

www.fs2b.ru

Тел. + 7 812 335 20 55
Факс + 7 812 335 20 39
E-mail: info@fs2b.ru

LINUX FORMAT

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов

Алексей Федорчук
Свою первую (и последнюю) программу написал еще на Алголе.

CRUX — крестоносец идеи

На заре тысячелетия ряд дистростроителей, утомленных сложностью дружественных пользователю дистрибутивов, обратился к истокам – первозданной простоте, свойственной Slackware, но уже на новом витке истории и с учетом накопленного опыта. На этой волне появились, почти одновременно, Gentoo, CRUX и Archlinux.

Судьба их оказалась различной. Gentoo, став самым популярным дистрибутивом в семействе Source Based, оказался центром большого и активного сообщества. Arch, развиваясь в направлении все большей пакетизации, приобрел менее широкий, но устойчивый и все более разрастающийся круг пользователей. CRUX же, занимающий промежуточное положение (и во многом послуживший прототипом для Arch'а), по сей день остается мало известным. Релизы его выходят почти с годичным интервалом. Выход последнего (2.3, 20 марта 2007 года) и послужил поводом для этой колонки.

CRUX распространяется в виде образа CD размером в 200 МБ, содержащего прекомпилированную систему, включающую не только X, но и WindowMaker, и разворачиваемую за считанные минуты. Такая компактность достигается за счет урезания «балласта» – в том числе всей документации, кроме man-страниц. Однако, это – все: за пределами «базы» нет ни одного пакета, ни одного репозитория, только порты, посредством которых собираются все остальные приложения. Коллекция портов не поражает своим объемом, но всегда актуальна. Ну а чего в портах не найдется – тут уж «спасение утопающих» ... сами знаете чье дело: сочинить собственный порт нужной программы в CRUX очень просто. И кстати: субъективно это самый быстрый дистрибутив, который я когда-либо видел...

alv@posix.ru

Сегодня мы рассматриваем...

12 Nokia N800
N770 продемонстрировал нам Linux в новом свете. N800 закрепляет достигнутый успех, добавляя емкую батарею, шустрый процессор, web-камеру и кое-что еще **8/10**

14 HP DVD940e
Все мы знаем, что устройства для записи DVD всего лишь прожигают маленькие питы в специальном слое. А не могут ли они выжечь что-то более креативное? Оказывается, могут! **8/10**

15 K3b 1.0
Ну признайтесь же: вы всегда тайно считали, что K3b – это уже 1.0. В конце концов, он был стандартным средством записи DVD много лет. Как же случилось, что он достиг заветного рубежа только сейчас? **9/10**

16 Greenphone
Процессор XScale? Есть. Qt? Есть. Сенсорный экран QVGA приличных размеров? Есть. Разработчики Linux плавно переходят на мобильные телефоны – благодаря Trolltech и Qtopia **8/10**

17 Metisse
Соперничество между Xgl и AIGLX набирает обороты, и Франция выставляет собственного претендента. Он не только полезен в работе, но и доступен в последнем Mandriva Live **8/10**

18 Sabayon Linux 3.2
Изысканный итальянский десерт с начинкой из Gentoo – убойный коктейль Linux-вкусностей **8/10**

Greenphone **с. 16**

› Да, на этот телефон можно зайти Telnet'ом – прямо из Норвегии. Что еще от жизни нужно?

Metisse **с. 17**

› *Metisse* сует свой нос в дела *Xgl* и *AIGLX*. Смотрите в Mandriva.

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.

Рейтинг 9/10

Nokia N800

Перед вами последний «интернет-планшет» от Nokia, с Linux на борту и массой новинок. **Ник Вейч** сует в него нос.

Вкратце...

» Мобильное устройство для выхода в Интернет, продолжает дело N770. Если вам нужен мобильный Linux, можно также «перепрограммировать» iPAQ.

Когда в 2005 году Nokia выпустила N770 – маленький черный кирпичик с Linux и сенсорным экраном (и почти ничего кроме) – многие решили, что это просто дорогая игрушка. Ведь специальность Nokia – сотовые телефоны, а не мобильные компьютеры.

N770 никогда не предназначался для массового рынка. Это была игрушка для хакеров и «proof of concept» (доказательство справедливости концепции). Но N770, видимо, и вправду что-то доказал, поскольку в этом году Nokia предъявила совершенно новую модель, N800, буквально лопающуюся от усовершенствований.

› **Интернет в вашем кармане, и не только. Но не пытайтесь с него звонить!**

Он – не телефон

Про новую серию устройств все хором задают один вопрос: почему их нельзя использовать как телефон? Что же в Nokia, не придумали, как его встроить? Ну, захотели бы – нашли бы способ. Но новое устройство предназначено отнюдь не для телефонных звонков: это просто мобильный выход в Интернет.

Внутренний Wi-Fi адаптер (b/g) позволяет войти в Сеть в зоне любой точки доступа. Если не получилось, подцепитесь через Bluetooth к вашему мобильнику (правда, если вы находитесь в Великобритании и в курсе тарифов на GPRS, вы крепко подумаете, прежде чем это сделать). ПО прекрасно справляется с задачей: утилита управления соединениями найдет ближайшие точки доступа и покажет вам их список с иконками, показывающими, кто из них защищен WEP, а кто нет (для подключения к защищенной сети понадобится указать секретный ключ). Подсоединившись, запускайте браузер – и вперед! Подключение к телефону или Wi-Fi происходит вполне непринужденно, причем информацию о соединении можно сохранять и не возиться с ключом каждый раз.

› **Вы не ограничены приложениями, поставляемыми Nokia: на www.maemo.org их гораздо больше.**

Вас расхолаживают воспоминания о потугах выйти в Интернет с других мобильных устройств? Забудьте: в этот раз все работает как надо. Модифицированная версия web-браузера *Opera 8* отлично выполняет рендеринг страниц, превращая даже самые сложные и неряшливо сделанные сайты в нечто удобочитаемое и простое для навигации. Прекрасный экран показывает все цветовые оттенки и имеет большие углы обзора плюс разрешение 800x480 точек, благодаря которому на этом экране все кажется больше.

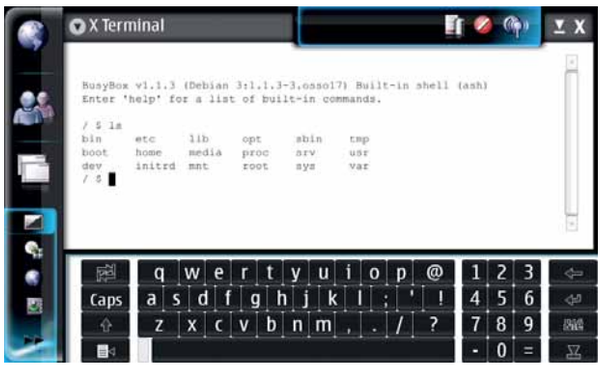
У N800 есть программа для чтения RSS-лент, клиент электронной почты, видео-проигрыватель и IM-пейджер, а в состав Opera включен Flash player. N770 не был плох, но его ядро с частотой 220 МГц заменили на 330-мегагерцевого монстра, и благодаря рывку процессорной мощности N800 удобен даже при работе с сайтами, богатыми графикой или Flash-анимацией. Конечно, отсюда не следует, что правильно отображаются все сайты мира (к сожалению, сейчас на любую комбинацию браузера и платформы найдется свой сайт-помойка, который будет показан ужасно). Больше всего проблем порождают сайты с большим количеством всплывающих окон или построенные на неприличных объемах JavaScript. Так что было бы замечательно, если бы Nokia встроила в N800 расширение типа «ad-block» (блокировка рекламы), чтобы сохранить память, трафик и здравый рассудок.

Он – не камера

Очень приятное дополнение к N800 – web-камера, хотя она и поддерживает только VGA-качество. Кто-то может счесть это просто игрушкой, но поддержка видео-связи и онлайн-конференций безусловно полезна, и вы вряд ли захотели бы тратить ресурс батарей на более мощную камеру с высоким разрешением. Странно, конечно, что в числе имеющихся приложений отсутствует программа записи видео. Несомненно, сторонние фирмы

Спецификация

Процессор: 320 МГц ARM11
Оперативная память: 128 МБ
Постоянная память: 256 МБ Flash
Экран: 4.1-дюймовый сенсорный
Разрешение: 800x480, 16-bit
Камера: 640x480 VGA
Связь: 802.11b/g Wi-Fi, Bluetooth 2.0
Разъемы: USB 2, наушники, питание, 2xSD
Размер: 144x75x18мм
Вес: 206 грамм



Установите оболочку – и найдете внутри вкусное ядрышко…

Сделай сам

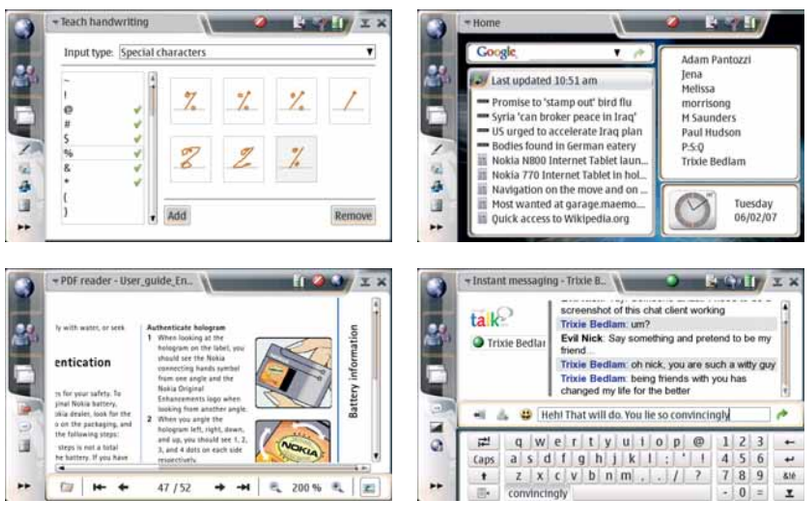
Не работай N800 под управлением Linux, не попал бы он на страницы журнала. И, вероятно, незачем было бы запускать на своем устройстве Linux, кабы не наличие общедоступных SDK и средств разработки, позволяющее настраивать и переделывать программы по своему вкусу.

Платформа разработки программ для интернет-планшетов Nokia существует с 2005 г. и называется *Maemo*. Имеющийся в ее составе SDK очень упрощает портирование приложений на платформу *Maemo* (фактически, вариант Linux на базе Debian), так что не удивляйтесь, если спустя месяц после выхода устройства вы увидите дюжины перенесенных из Linux программ. Среди программ третьих фирм, уже имеющихся на N770, есть игры, файловые менеджеры, оболочка *Bash*, астрономические приложения и так далее. Сейчас многие из них перенесены и на N800 (для реализации некоторых новых функций N800, например, камеры, потребовалось обновление ОС).

моментально запланируют ее реализацию, но результат может оказаться не так уж хорош – сжатия, позволяющего эффективно использовать имеющееся дисковое пространство, добиться достаточно тяжело.

Зато можно смотреть фильмы: хотите – зайдите на YouTube, хотите – загрузите файлы в одном из множества форматов, поддерживаемых видеоплейером. Конечно, качество будет похуже, чем на вашем двухпроцессорном ноутбуке – фильмы иногда дергаются, бывает, что теряются кадры и возникают проблемы синхронизации со звуком. Короче, вы вряд ли захотите смотреть на N800 последний блокбастер. Однако встроенный проигрыватель в изобилии поддерживает форматы (RealAudio, MP3, AAC, WMA, WAV, M4A), и, естественно, при наличии сетевого подключения вы можете слушать потоковое аудио с ваших любимых радиосайтов.

Вероятно, в id software, представляли не совсем такое, программируя свою великую стрелялку, но играть в *Doom* вполне можно.



ПО позволяет общаться, сидеть в Интернете, писать и читать письма. Отлично распознается рукописный ввод, и экранной клавиатурой тоже можно пользоваться, даже пальцами.

При такой активности батареям N800 есть о чем подумать, однако наши тесты показали, что они легко выдерживают четыре-пять часов работы средней интенсивности и несколько дней в режиме ожидания.

Он – не для всех

К сожалению, N800 остается по большей части устройством для технарей. Хотя его 7 на 14 (примерно) сантиметров неплохо умещаются в большой карман, большинство людей предпочтет захватить свои ноутбуки. И, честно говоря, пока цены на сотовую связь так несуразно завышены, нельзя считать, что N800 «всегда в сети» – кроме как для тех, кто не дрогнет перед стоимостью подключения (примерно 1 фунт за мегабайт без каких бы то ни было скидок [в Великобритании; в России, как ни странно, стоимость GPRS-соединения приемлемая, – прим. перев.]).

Но разве не здорово подключаться к рабочему столу вашего домашнего компьютера, сидя за столиком придорожного кафе? Мы думаем, достаточно здорово, чтобы соблазнить пару-тройку будущих покупателей N800. LXF

Ник вспоминает N770

«N770 был скорее публичным прототипом, чем моделью для массового рынка, с простой (хотя вполне функциональной) панелью, с ограниченной мощностью и странным угловатым корпусом. Но, попользовавшись им месяцев шесть, я его полюбил. Его металлический корпус оказался достаточно прочным, чтобы пережить соседство в кармане с ключами и швыряние в сумку в спешке перед поездом.

Кнопки N770 не сказать чтобы особенно элегантные, но зато ими очень удобно пользоваться, даже в темноте. Мне нравилось, что он напоминал большой прямоугольный кирпич. Экран работал прекрасно, стилус было немножко менее удобным, но вполне пригодным к использованию там, где не справлялись пальцы.

Хотя добавка стерео-динамиков в N800 приятна, камера полезна, а время жизни на батареях очень солидное, мне кажется, что на пути к нему что-то было утеряно. Для Nokia N800 ближе к «мэйнстриму», но выполняя опасные миссии на службе *Linux Format*, я прихвачу с собой свой N770.»

LINUX FORMAT Вердикт

Nokia N800

Разработчик: Nokia
Сайт: www.nokia.com
Цена: $399

Функциональность	8/10
Производительность	9/10
Простота использования	8/10
Цена	8/10

Вероятно, это пока скорее игрушка для хакеров, чем жизненно необходимое устройство, но она заслуживает высокой оценки.

Рейтинг 8/10

HP DVD940e

Устройство, записывающее и подписывающее диски? В Linux? **Ника Вейча** прожгла мысль: а оно нам надо?

Вкратце...

» DVD-привод с поддержкой технологии LightScribe, умеет прожигать метки на специальных дисках. См. также модели от Philips, LaCie, Samsung и LG.

Нечасто мы обсуждаем на этих страницах вещи столь простые, как пишущий DVD-привод, но этот привод не совсем обычный. DVD940e от HP – это устройство с поддержкой технологии *LightScribe*. С ее помощью можно наносить рисунки на покрытие нерабочей поверхности особых дисков, которые в результате выглядят круто и/или профессионально. Технология известна довольно давно, но по некоторым причинам используемые ею протоколы были тщательно засекречены. Теперь HP позаботилась, чтобы Linux-пользователи извлекали максимум пользы из своего оборудования, и выпустила для разработчиков Linux-библиотеки под закрытой лицензией. В настоящее время пользовательское ПО от LaCie и Lite-On позволяет наносить рисунки на диски в вашей любимой ОС.

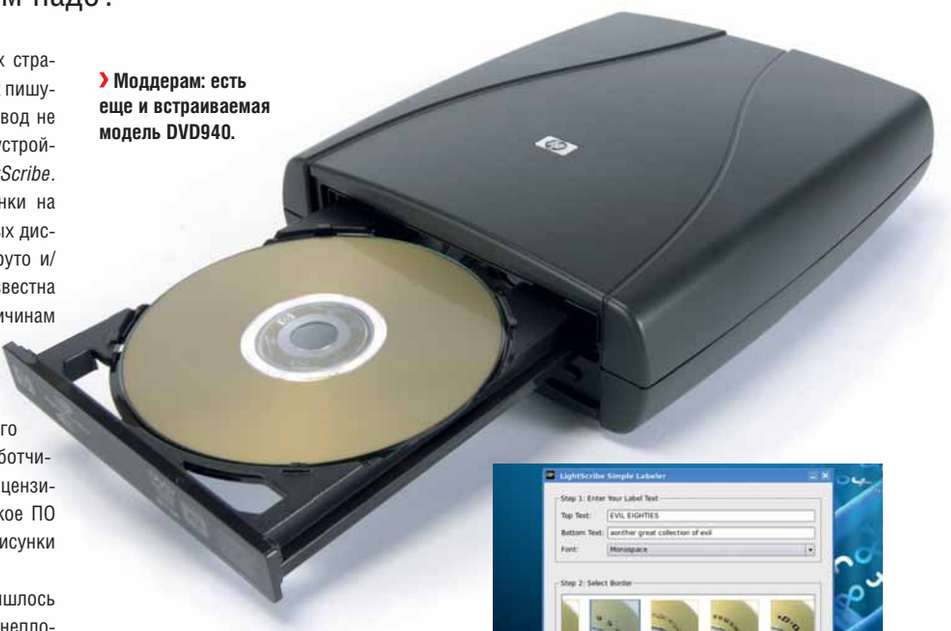
После инсталляции ПО нам-таки пришлось почесать затылки. Хотя выглядело это неплохо, обе рисовальные программы заявили, что не видят устройства (*K3b* его прекрасно видела). Кажется, проблема заключалась в библиотеках – небольшое расследование показало, что их смутил поиск по всем доступным устройствам. Это случилось на Fedora, а затем повторилось на Ubuntu – а вот SUSE все опознал верно, поэтому ваш путь к успеху может быть разной длины.

Доступны два варианта: *Simple Labeller* от Lite-On и *4L LightScribe Labeler* от LaCie, оба поставляются как RPM для платформы x86. Также планируется ввести поддержку *LightScribe* в *K3b*, что определенно сократило бы количество проблем при выигрыше в функциональности. Решение от Lite-On восхитительно, но ограничивается текстом и коекакой встроенной графикой. Программа LaCie, наоборот, умеет наносить рисунки, но не способна воспроизводить текст. Независимо от выбранной программы, прожиг этикетки на нашем приводе занял 22 минуты! Если это чересчур долго, обе программы предлагают нанести лишь отдельные участки рисунка (например, узкое кольцо простого текста): тут потребуется меньше пяти минут.

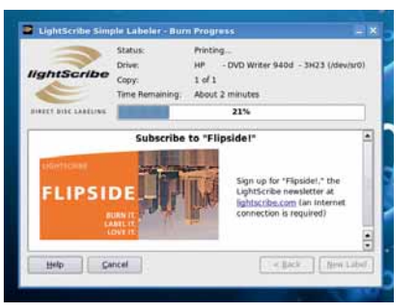
Что касается самого привода, то он вполне соответствует нынешнему времени – записывает DVD+ и DVD- на 18x, двухслойные – на 8x. DVD940 поддерживает DVD-RW, DVD-RAM и, естественно, CD-R (на 48x). Есть кнопка, есть лампочка – чего вам еще?

Единственный минус *LightScribe* – цена. Нет, сам дисковод не дороже обычного, а вот специальные диски стоят раза в два больше. И все же, если вы желаете произвести впечатление этикетками, теперь это возможно и под Linux. Надеемся, HP вскоре выпустит библиотеки под GPL и устранит проблемы с опознанием оборудования. LXF

› **Для улучшения результатов работы с рисунками в программе от LaCie поработайте с контрастом.**



› **Моддерам: есть еще и встраиваемая модель DVD940.**



› ***Simple Labeller*** **умеет рисовать только текст и несколько видов рамок, но выглядит это замечательно.**

LINUX FORMAT Вердикт

HP DVD940e

Разработчик: Hewlett Packard
Сайт: www.hp.com
Цена: $129.99

Функциональность	8/10
Производительность	8/10
Простота использования	6/10
Документация	8/10

» *Хороший привод, а ПО хоть и неприятзательное, но вполне работоспособное.*

Рейтинг 8/10

K3b 1.0

Постоянные читатели знают: **Майк Сондерс** прожигает уйму дисков в огнедышащих печах Башен LXF. Кто же лучше него проверит K3b на соответствие стандартам?

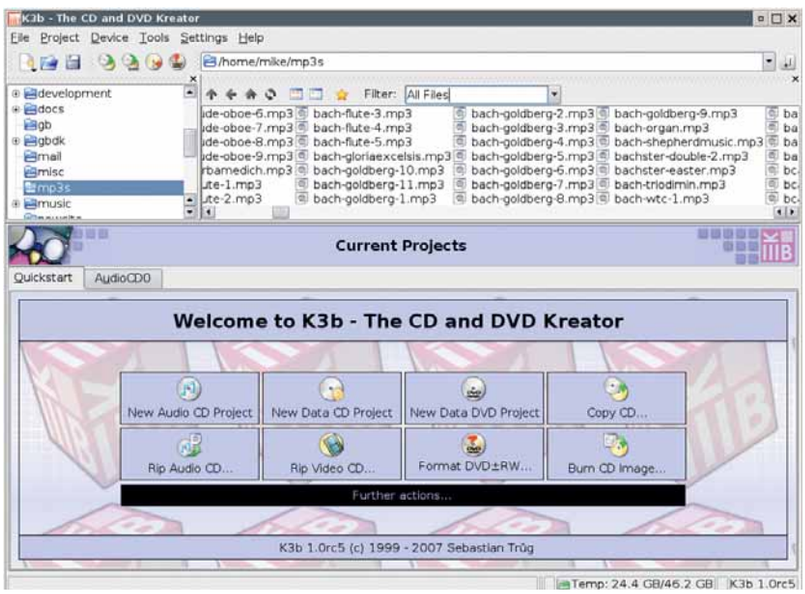
Вкратце...

» Программа записи CD/DVD для KDE. См. также: GnomeBaker или X CD-Roast.

В последние годы *K3b* продвигалась прогулочным шагом – с ноября 2003 по июнь 2006 имели место лишь мелкие подвижки, от 0.10.3 до 0.12.6, в основном по причине достаточности дискозаписывающего арсенала KDE, не дававшей повода к большим скачкам. Теперь же, по мнению ведущих разработчиков Себастьяна Трюга [Sebastian Trüg] и Кристиана Квасны [Christian Kvasny], стабильность и лоск приложения наконец заслужили «звание» 1.0. Верно ли выбрано время выпуска первой полной версии?

K3b компилируется и инсталлируется используя лишь стандартные библиотеки KDE, хотя для дополнительной функциональности может понадобиться *Transcode* (чтение DVD), *FFmpeg* (чтение видео) и *FLAC* (для поддержки сжатия без потерь). Внимание к деталям заметно уже на старте, когда *K3b* проверяет целостность файлов, актуальность установленных программ и подробно описывает результаты проверки в диалоговом окне.

«Отлаженное управление медиа-форматами вывело программу в передовики.»



› У K3b есть даже собственные темы. Здесь – пингвин с размытой тенью.

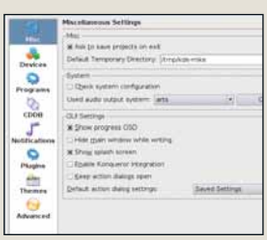
По сравнению с *GnomeBaker*, интерфейс *K3b* переполнен значками, списками файлов и панелями обратной связи, разбросанными по всему главному окну (*см. врезку «K3b против GnomeBaker» слева*). Но удобный стартовый экран предоставляет доступ к большинству стандартных функций записи CD: созданию аудиодисков или CD/DVD с данными, а также копированию CD за один щелчок. Запись дисков гениально проста – просто перетаскиваете файлы и каталоги из файлового навигатора на панель проекта. Благодаря вкладочному интерфейсу можно работать над несколькими проектами записи одновременно (собирая, например, музыкальный диск и диск с данными). Кроме того, *K3b* умеет непосредственно копировать диски и записывать ISO-образы.

Без усилий

Откровенно говоря, все это стандартно для любого ПО, записывающего CD, а вот отлаженное управление разнообразными медиа-форматами вывело *K3b* из общей массы в передовики. Программа может конвертировать файлы MP3, OGG и WMA в аудиоконтент, а также копировать CD в MP3 и OGG. На видеофронте программа умеет превращать DVD в форматы DivX и Xvid (с помощью *Transcode*), а также записывать менее популярные VCD (1.1, 2.0 и SVCD). Таким образом, *K3b* не просто механическая «прожигалка», а полный комплект для записи, чтения и конверсии мультимедиа.

Если у вас установлена версия *K3b* 0.12.x, обновиться явно стоит: в 1.0 входит Video DVD KIO slave для конвертирования DVD через *Konqueror*, плюс усовершенствованное управление устройствами (размонтирование и извлечение дисков теперь возможно из меню *K3b*). URL медиа-устройства можно набрать в командной строке при запуске *K3b*, а в течение записи или конвертирования отображается оценка оставшегося времени. Без сомнения, *K3b 1.0* на сегодняшний день наиболее полный комплект для записи и извлечения дискового контента под Linux, к тому же очень надежный благодаря зрелой кодовой базе. LXF

K3b против GnomeBaker



› В типичном для KDE стиле, *K3b* предоставляет гораздо больше возможностей настройки, чем Gnome-аналог.

› Эй, *GnomeBaker*, не лучше ли вывести сообщения об ошибках, вместо корявого журнала записи диска?

GnomeBaker – основной инструмент для записи дисков в Gnome, но до *K3b* ему далеко. Милый фасад опрятного и дружелюбного интерфейса в случае малейшей неудачи сменяется жутким техноязом сообщений об ошибках.

А главное, у *GnomeBaker* нет средств извлечения или конверсии мультимедиа: это просто «прожигалка». Вознамерившимся «содрать» с диска кое-какие данные придется собирать кучу сторонних приложений. И если вы поклонник Gnome, утомленный скудостью *GnomeBaker* – попробуйте *K3b*, оно отлично работает на вашем любимом рабочем столе.

LINUX FORMAT Вердикт

K3b 1.0

Разработчик: Команда K3b
Сайт: www.k3b.org
Цена: Бесплатно под GPL

Функциональность	10/10
Производительность	8/10
Простота использования	8/10
Документация	7/10

» *K3b набита почти всеми мыслимыми функциями для записи дисков и снабжена блестящей GUI-оболочкой.*

Рейтинг 9/10

Greenphone

Это не простой телефон: он построен на открытом ПО. Пробует **Пол Хадсон**.

Вкратце...

» Мобильный телефон на открытом ПО, с неограниченным хакерским потенциалом. См. также: Motorola A780 и подобные.

Грэм считает...

«Идея открытого ПО благородна, но ведь не любой ценой! Пользоваться Java на своем телефоне гораздо дешевле, а функциональность от этого не пострадает.»

Начальные вопросы снимаю сразу. Да, Linux таки работает на мобильных телефонах. Нет, грузится нисколько не быстрее. Нет, эту трубку вы никогда не увидите в магазине за углом, и – да, телефон действительно пронизан компонентами открытого ПО.

Greenphone предназначен только для разработчиков и призван показать, что Qtopia – подходящий компонент для крутой мобильной ОС. Если вы когда-либо собирали приложения *Qt* или даже KDE, это – кратчайший путь к освоению мобильного программирования. Даже если вы не *Qt*-программист, Greenphone – простейшее средство крутого хакерства на Linux-основе. Но – не без недостатков…

Greenphone – не для всех. Хотя общее качество изготовления исключительно хорошее, стилус для 2.5-дюймового QVGA-экрана при извлечении царапает внутреннюю поверхность, Bluetooth слабоват, да и ПО далеко от завершения. Но разве это не круто – программируемый телефон? Поэтому прилагается USB-кабель для зарядки и синхронизации с компьютером, а SDK (Software Development Kit, арсенал программиста) полон инструментов.

Сам SDK выполнен довольно грамотно: он предлагается в виде образа для *VMware*. На Linux нужно лишь установить бесплатный пакет *VMware Player* – и дело с концом. Windows-пользователи тоже не обижены, так как SDK запустится на любой платформе, на которой работает *VMware Player*. Виртуальная машина (ВМ), в которой запускается SDK – это вариант Debian, модифицированный для загрузки прямо в KDE и настроенный так, чтобы все инструменты программиста Qtopia были под рукой. Преднастроено и соединение с Greenphone, то есть подсоединив USB-кабель, вы можете подключаться к телефону через встроенную подсистему «Ethernet поверх USB» Да, в телефоне есть Telnet – и BusyBox с ядром Linux 2.4.19, так что вы легко можете получить доступ с правами root.

Хотя совсем нетрудно перемещать приложения на Greenphone через USB-кабель или встроенный слот для чтения карт Mini SD, все же лучше пользоваться для тестирования кода встроенным [в SDK, – прим. ред.] эмулятором, а затем с помощью кросс-компилятора создавать специфические Greenphone-версии. Есть даже система удаленной отладки, позволяющая собирать, инсталлировать и запускать приложения на Greenphone, управляя ими из ВМ SDK. Все это осуществляется через отладчик Kdbg и USB и работает безупречно: можно запустить приложение из SDK, а затем в любое время обратиться к его GDB-серверу.

Железная сторона

Убедившись, что Greenphone действительно может удовлетворить любую прихоть разработчика, перейдем к оборотной стороне: телефон-то слабоват. Внутри – процессор XScale 312 МГц, 64 МБ ОЗУ и 128 МБ флэш-памяти, 1,3-мегапиксельная камера, плюс обычная связь GSM/GPRS. Сенсорный экран – это мило, но сам телефон очень медленный: не меньше 38 секунд на включение и 2-3 секунды на запуск любого приложения – будь то часы или список контактов. Удручает, что приложения не грузятся быстрее при повторном запуске – вероятно, потому, что общий вес Linux + *Qt* оставляет на все остальное лишь 8 МБ из имеющихся 64.

Учитывая то, что трубка создана лишь для разработки ПО, ни одна из этих проблем не выглядит серьезной. Greenphone – ошеломляюще простой путь для переноса *Qt*-приложений на мобильную основу, и его возможности прямо-таки хлещут через край. Наконец-то появилась воистину открытая платформа для разработки мобильного ПО, в противовес iPhone от Apple. Только представьте: изготовители телефонов обращают внимание на ПО Greenphone, дорабатывают его, а затем распространяют в своих изделиях, конкурируя с самой Symbian! Признаться, Greenphone откровенно слаб, если все, что вам нужно – это Linux в кармане; а вот Motorola уже «окучивает» этот рынок, и на ее телефонах Qtopia тоже работает! LXF

› **Если GPL-программирование – не для вас, за лишние $195 вы получите закрытую лицензию на одно устройство.**

LINUX FORMAT Вердикт

Greenphone

Разработчик: Trolltech
Сайт: www.trolltech.com
Цена: $695 за трубку + SDK под GPL

Функциональность	7/10
Производительность	6/10
Простота использования	10/10
Цена	7/10

» *Неплохо для хакера, еще лучше для серьезной разработки, но только не для домашнего использования.*

Рейтинг 8/10

Metisse

Надоела истерия вокруг дрожащих окон и вращающихся кубов? И **Нейлу Ботвику** тоже; он занялся 3D-рабочим столом, для которого главное – функции.

Вкратце...

» Трехмерный рабочий стол и инструментарий, продукт исследования интерфейса пользователя In Situ. См. также: *Xgl*, *AIGLX*.

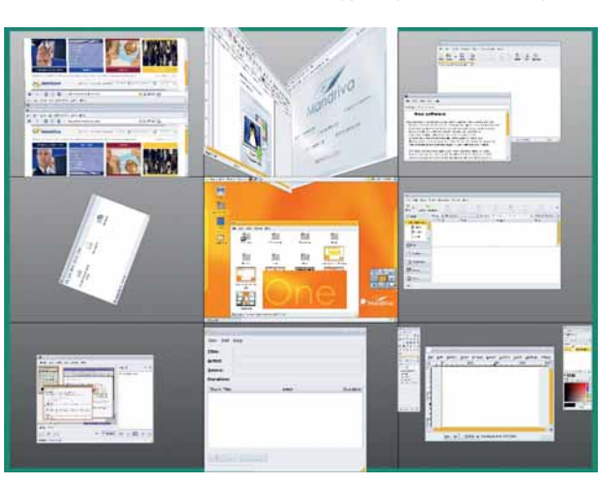
Трехмерные рабочие столы – вроде автобусов: то ждешь их целую вечность, то они приходят по три штуки сразу. *Xgl* и *AIGLX* последнее время постоянно на слуху, а тут и команда In Situ выдала нам *Metisse*, инструментарий под GPL, способный наделить объемом системы, которые «не тянут» вращающиеся кубы и дрожащие окна.

На первый взгляд *Metisse* похож на прочие попытки придать объем рабочему столу, только скромнее на вид. Окна не дрожат и не мерцают, но их можно вращать – не только в плоскости экрана, но и «вглубь»: за счет некоторой потери читаемости эффективнее используется экранное пространство. При движении окна становятся прозрачными, а если понадобился доступ к данным под ними – отгибаются.

Движимость

Вместо размещения нескольких рабочих столов на кубе Metisse использует раскладку на странице: отображение девяти столов в сетке «три на три». В этом режиме можно увеличивать любой из столов и даже перетаскивать окна из одного стола в другой. В отличие от аналогов, окна в *Metisse*, даже будучи трансформированными, по-прежнему работоспособны: все меню и экранные элементы управления функционируют, как ни в чем не бывало. Нечто подобное планируется осуществить в будущих версиях *Compiz*, а с *Metisse* возможно уже сегодня.

Основная цель трансформации окон – наиболее выгодное использование экранного пространства, которого вечно не хватает при большом числе открытых окон. Можно вращать окна вокруг вертикальной и горизонтальной осей, перемещать между рабочими столами и даже использовать для соединения столов. Любой рабочий стол может также перекрывать собой часть соседних. Это чем-то напоминает виртуальный рабочий стол, только нет автоматической прокрутки при достижении границ. Замечательно, что окна можно просматривать и взаимодействовать с ними на всех девяти страницах разом. Отдельные окна можно увеличить, а можно сделать прозрачными, и все они становятся прозрачными при перемещениях. Трансформацию можно отменить нажатием клавиш; можно также временно сдвинуть окно и посмотреть, что под ним – оно вернется назад, когда клавиша будет отпущена.



› **Metisse дает намного больше власти над окнами, чем обычные экранные значки.**

› **Хотя Metisse выпускается под GPL и открыт для всех, разработчикам помогает Mandriva: они поместили этот инструментарий на новейший Live CD Mandriva One.**

Все это управляется смесью клавиатурных и «мышиных» команд. Например, для поворота окна вокруг вертикальной оси можно выбрать пункт меню и протащить указатель мыши либо нажать F4 или F5 в сочетании с клавишей Windows (клавиша Linux на клавиатуре Cherry).

Мы пробовали *Metisse* на Live CD Mandriva One, который использует Gnome, однако *Metisse* непривередлив к рабочим столам и будет работать в большинстве из них – хотя, например, пользователи *Ratpoison* вряд ли им заинтересуются. *Metisse* базируется на FVWM, одном из самых гибких оконных менеджеров, и Mandriva планирует включить его в свою следующую версию своего дистрибутива. Но *Metisse* – свободная программа, и может быть включена в любой дистрибутив (уже имеются сборки под Gentoo).

Вращающиеся кубы с 3D-ускорением слишком часто наводят на мысль, что это красивый способ работать как можно медленнее. *Metisse* идет другим путем: меньше пыли в глаза, больше удобства в работе. Мы уверены в том, что программа войдет и в другие дистрибутивы, и спасибо Mandriva за то, что сумела ее разглядеть и представить всеобщему вниманию. LXF

LINUX FORMAT Вердикт

Metisse

Разработчик: Оливье Шапюи и Николя Руссель
Сайт: http://insitu.lri.fr/metisse
Цена: Бесплатно под GPL

Функциональность	9/10
Производительность	8/10
Простота использования	8/10
Документация	6/10

» *Наконец-то рабочий стол, который использует 3D-технологии для повышения продуктивности.*

Рейтинг 8/10

Sabayon Linux

Итальянский дистрибутив Sabayon Linux уже больше полугода цепко удерживается в первой десятке на Distrowatch.com, наступая на пятки Mandriva, однако на российском рынке он пока не слишком заметен. Прикинувшись итальянцем, **Сергей Супрунов** решил взглянуть на него поближе...

Вкратце...

» Основные дистрибутивы Linux предпочитают придерживаться здорового консерватизма в вопросе следования последним тенденциям. Sabayon же предлагает вам увидеть все и «здесь и сейчас»! См. также: Kororaa и Kororaa LiveCD

Sabayon – не такой уж и молодой дистрибутив, просто раньше он носил странное имя RR (в ходу были две версии – RR4 для платформ x86 и RR64 для 64-разрядных архитектур). Впрочем, имя Sabayon (это название итальянского десерта) кому-то может показаться не менее странным, так что на своей Wiki-страничке разработчики первым делом объясняют, что к чему.

В основе дистрибутива лежит Gentoo Linux, так что можно рассчитывать на всю мощь коллекции Портежей (Portage) и преимущества индивидуальной сборки. Хотя, честно говоря, запрятаны эти возможности настолько глубоко, что поначалу Sabayon вполне можно принять за «чисто пользовательский» дистрибутив.

На старт!

Sabayon Linux представляет собой LiveDVD, так что его можно испытать прямо «с колес», не утруждая себя поисками хотя бы узкой свободной полоски на диске между Ubuntu и FreeBSD (да еще этот NTFS-раздел с ценными данными, будь он неладен...). Рассматриваемая здесь версия – 3.26 для платформ x86, но нужно сразу отметить, что Sabayon развивается просто нечеловеческими темпами, стремясь угнаться за всеми последними новинками даже не сезона, а недели.

Так, версия 3.2 увидела свет в начале декабря 2006 года. Уже 2 января 2007-го (и праздники им там не помеха) появился релиз 3.25, а всего через шесть дней – 3.26. В тот самый момент, когда делался этот обзор, уже был анонсирован 3.3, так что к тому времени, когда вы будете читать эти строки, уже наверняка появится еще что-то новенькое.

Ключевое ПО Sabayon

- Ядро 2.6.19-gentoo-r4
- X.org 7.2rc3
- KDE 3.5.5
- Gnome 2.14
- Beryl 0.1.4
- GCC 4.1.1
- glibc 2.5
- Firefox 2.0
- Thunderbird 1.5.0.7
- OpenOffice.org 2.0.4
- Gimp 2.3

Итак, после загрузки с DVD перед вами предстанет знакомое «джентльменам» меню выбора дальнейшего действия (загрузка, инсталляция, установка в текстовом режиме и т.д.), однако присутствует здесь и пара новых пунктов: «Start in Internet Kiosk NX mode» и «Start Quake 4 demo», с помощью которых без лишних слов можно сразу запустить нужное в сессии Fluxbox. Внизу экрана можно выбрать язык (есть и русский), но, честно говоря, первой мыслью было, что это «пережиток» Gentoo и на дальнейшей работе никак не скажется (на сайте была заявлена поддержка только английского, испанского, французского и, само собой, итальянского). К нашей всеобщей радости, русификация оказалась выполненной на довольно высоком уровне.

До появления на экране менеджера KDM вам предстоит сделать еще один выбор – включать ли поддержку трехмерного рабочего стола (причем предлагается как AIGLX, так и XGL; да, чуть не забыл – драйверы от производителей уже установлены, так что вам с этим возиться не придется). После чего – добро пожаловать в систему (пользователь – sabayonuser, пароль пустой)!

Рабочий стол по умолчанию – KDE, хотя с первого взгляда его трудно узнать – две «гномьи» панели, нестандартное главное меню... Если ваша видеокарта достаточно крута, то «прямо из коробки» вы получите все прелести, которые способен предложить *Beryl* – и вращающийся куб, и желеобразные окна, и потрясающие эффекты сворачивания/разворачивания окон, и их полупрозрачные заголовки, и даже ливень, вызываемый по Shift-F9.

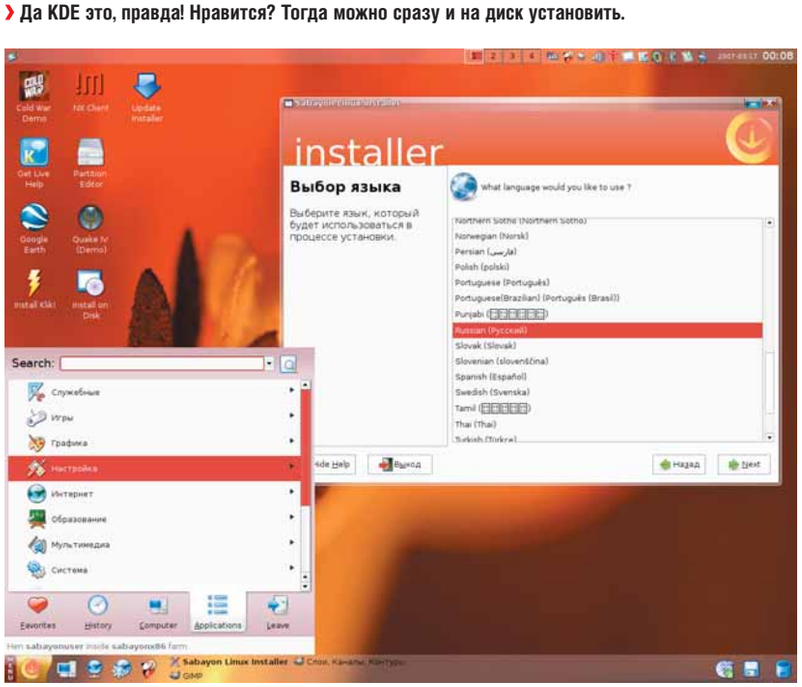
Интересно реализовано главное меню – несколько вкладок, группирующих основные элементы управления, и строка мгновенного поиска – результаты будут отображаться и уточняться сразу же по мере ввода вашего запроса (спасибо *Beagle*).

Ну а найти в этом меню (на вкладке «Applications») можно почти все – офисные пакеты *KOffice* и *OpenOffice.org*, *AbiWord*, всевозможные графические и издательские инструменты (*Gimp 2.3*, *Inkscape*, *Blender*, *Xara*...), пакеты разработчика (*KDevelop*, *Anjuta*, *GLADE*, *Quanta Plus*...), массу интернет-инструментов (*Firefox/Thunderbird*, *Google Earth*, *Azureus*, *Opera*, *SeaMonkey*, *Skype*, *KTorrent*, *OpenVPN*...). В общем, создается впечатление, что разработчики решили пойти по пути «Мы дадим вам все, что есть, а лишнее сами выкинете».

Мультимедиа тоже на высшем уровне – с десяток различных плейеров, поддержка «из коробки» MP3, видео... Для все еще сомневающихся в возможностях Linux – демо-версии *«Quake IV»* и *«Cold War»* (первую можно запустить сразу из загрузочного меню диска).

Помимо KDE, щелкнув в KDM на «Тип сеанса», можно выбрать Gnome, Xfce4, Fluxbox,

› Да KDE это, правда! Нравится? Тогда можно сразу и на диск установить.



Enlightenment. Хотя «трехмерностью», понятное дело, насладиться можно только в Gnome.

А что же знаменитая мощь и гибкость администрирования, присущая Gentoo? К сожалению, графическими средствами настройки она не обросла, оставив вас с традиционными инструментами Центра Управления KDE. Для управления пакетами есть графический интерфейс к системе Портежей – *Kuroo* – однако без жесткого диска поработать с ним не получилось.

Но не забывайте, что у вас есть консоль! Вся мощь команды *emerge* к вашим услугам, так что если клавиатура вас не пугает, ознакомьтесь с документацией на www.sabayonlinux.org/wiki, и вперед!

Внимание!

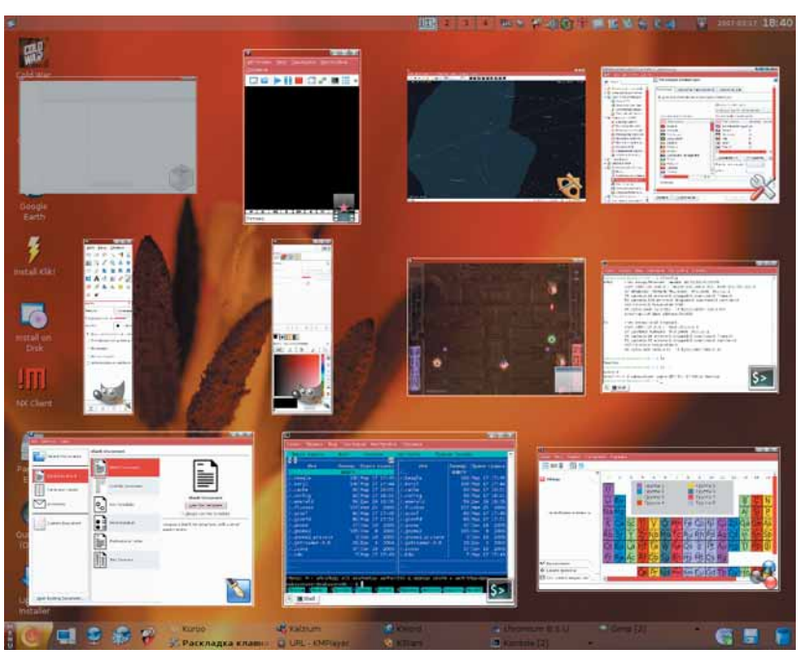
Теперь – про замеченные (хотя, наверное, и спорные) недостатки. Прежде всего, любителей Gnome очень огорчит пренебрежительное отношение разработчиков к этому рабочему столу. Уж не хочется возиться с настройками – оставьте все по умолчанию... Так нет же – одна панель вместо традиционных двух, отсутствие основного меню, да еще и на рабочий стол вывалили все иконки найденных разделов. Прямо Knoppix какой-то... Но, опять-таки, нужно отметить нетрадиционный подход к меню – та же строка поиска, что и в KDE, а для непосредственного доступа к приложениям следует нажать кнопку «More Applications...», открывающую отдельную папку с иконками всех установленных программ.

Вообще, в Sabayon частенько натыкаешься на отклонения от «традиций». То ли разработчикам нравится быть не такими, как все, то ли действительно мы привыкли жить неправильно, а истинное удобство – вот оно, нужно только сделать первый шаг и отказаться от въевшихся с годами привычек... Как бы то ни было, в Sabayon постоянно чему-то удивляешься: KDE с двумя панелями больше напоминает Gnome (и даже начинает нравиться), Gnome же, наоборот, скорее похож на KDE... Только вы захотите нажать на кнопку «Menu», как она сама распахивается у вас перед носом; зато подведя мышку к одному из пунктов этого меню, можно ждать целую вечность, пока раскроется подменю – здесь, оказывается, требуется явный щелчок...

› Даже Beryl рыдает над участью Gnome... Хотя определенный шарм во всем этом есть.

Переизбыток красного в «умолчальной» теме довольно быстро утомляет, полупрозрачные заголовки требуют дополнительных усилий при работе с окнами, не позволяя моментально определить границы окна. Впрочем, все это элементарно корректируется в настройках, хотя для LiveDVD мы бы предпочли более сдержанное оформление, принятое по умолчанию, иначе дистрибутив становится трудно применять на регулярной основе, и он так и остается в роли «демонстрашки». Впрочем, никто не запрещает поставить его и на диск – соответствующий пункт предлагается во время загрузки, да и прямо из загруженной в режиме LiveDVD системы можно также запустить Installer.

› Стоит лишь подвести мышь к правому верхнему углу безнадежно захламленного рабочего стола – и все ваши окна послушно выстроятся ровными рядами..



Как ни странно, но с первых же минут пришлось лезть в консоль – запустить *KNetworkManager* по непонятной причине не удалось, а другого способа «цивилизованно» настроить сеть не видно. Впрочем, если в вашей сети работает DHCP-сервер (что даже для домашнего использования в условиях широкого распространения ADSL уже не редкость), то с этим неудобством вы не столкнетесь.

Ну и еще одна досадная мелочь – если при загрузке выбрать русский язык, то в дальнейшем в переключателе раскладок у вас будет две русских и ни одной английской (обойти это можно, если после выбора языка нажать F3 и вернуть американскую раскладку). Впрочем, если предполагается поработать в Sabayon хотя бы час, визита в Центр Управления все равно не миновать...

Марш!

Что ж, первое впечатление от этого дистрибутива: «А Linux-то и правда крут!». Даже если вы и так не сомневаетесь в этом. Разработчики затолкали на 3 ГБ практически все, чем Linux может похвастаться на текущий момент, и то, что демонстрация возможностей системы ставится ими превыше идеологической демагогии об открытости и «чистоте», позволяет нам, простым пользователям, получить все эти прелести здесь и сейчас, без лишних усилий. LXF

LINUX FORMAT Вердикт

Sabayon Linux

Разработчик: Sabayon Linux
Сайт: www.sabayonlinux.org
Цена: бесплатно под GPL

Функциональность	9/10
Производительность	7/10
Простота использования	8/10
Документация	9/10

» Если оперативной памяти достаточно, то даже работа в режиме LiveDVD не помешает вам насладиться трехмерным рабочим столом и крутыми играми.

Рейтинг 8/10

Сравнение

Каждый месяц мы анализируем для вас тысячи программ — а вы можете отдохнуть!

LINUX FORMAT На DVD

Системы управления версиями

Git или *Bazaar*, *CVS* или *Subversion*: что лучше для вашего проекта? Выясняет **Грэм Моррисон**.

Paul Blachford

Наш выбор

Aegis с. 22
Bazaar с. 24
CVS с. 21
Git с. 24
Monotone с. 22
RCS с. 21
Subversion с. 23

Про наш тест

Мы рассмотрели семь наиболее солидных систем управления версиями, доступных для Linux. Есть несколько прекрасных коммерческих продуктов, но мы сосредоточились на открытых решениях. Отчасти потому, что они доступны свободно, но также и потому, что открытые системы обычно одни из лучших, независимо от цены. Мы рассматривали простоту использования, качество документации и производительность каждой системы, а также их прозрачность в работе. Победитель этого Сравнения должен быть мощным и не создавать препятствий процессу разработки.

Нет сомнения: без систем управления версиями Linux никогда не достиг бы такого расцвета. Управление версиями связывает тысячи разработчиков, собирая сделанное ими и предоставляя результаты остальным. Оно похоже на клей, удерживающий проект в целости и равновесии.

Все системы управления версиями используют один и тот же основной принцип. Они регистрируют изменения в программном коде – или, в правильной терминологии, отслеживают версии. Каждое изменение (обычно представляющее собой добавление нового файла или модификацию существующего) порождает новую версию. Другой важный аспект систем управления версиями – совместная работа. Они должны позволять разработчикам трудиться вместе и держать каждого из них в курсе изменений, сделанных товарищами.

Существуют два способа решения этой задачи. В более простом случае для размещения всех файлов проекта используется отдельный сервер. Затем каждый разработчик создает личную копию дерева исходного кода и локально применяет к ней свои изменения, а потом синхронизирует их с данными на сервере. Так работают *CVS* и *Subversion*, наиболее популярные системы управления версиями.

Путь Торвальдса

Другой подход к совместной работе скорее похож на передачу файлов в пиринговых (peer-to-peer) сетях, чем на разработку приложений. Вместо использования центрального сервера, хранящего все данные проекта, каждый разработчик отвечает за синхронизацию своей рабочей копии проекта с остальными разработчиками. Проект становится децентрализованным. Этот подход, судя по всему, является модной тенденцией, если принять во внимание *Git* Торвальдса и ряд других систем, успешно применяющих децентрализованное управление версиями.

Разработчики используют системы управления версиями для координации своих усилий при исправлении ошибок и написании новых фрагментов кода. Но система управления версиями не ограничивается программистами. Возможно, вам захочется воспользоваться клиентом для загрузки последней версии своего любимого приложения или для управления документацией, или даже для своей почты. Систему управления версиями не волнует тип файлов.

Если вы ищете себе систему управления версиями, сейчас самое время побольше узнать о существующих разновидностях. Это одна из областей Open Source, где коммерческие продукты сталкиваются с действительно серьезной конкуренцией, а основные инструменты становятся все лучше с каждым релизом.

RCS

Старейшая система в нашем Сравнении.

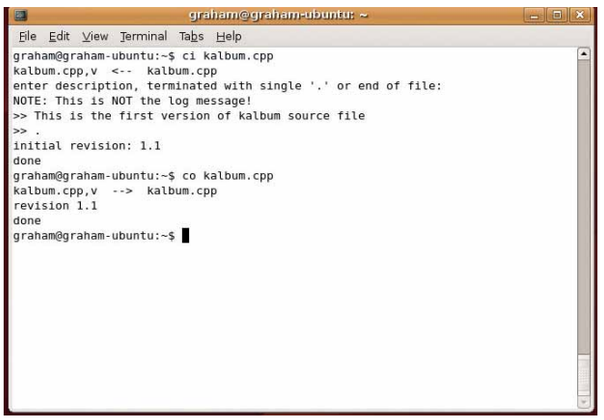
Исходная система управления пересмотрами версий (Revision Control System; да-да – *RCS* и вправду действует согласно этикетке), может, и вытесняется *CVS*, но остается популярным выбором в некоторых особых случаях, и есть две главных причины предпочесть *RCS* более продвинутым решениям. Во-первых, эта система сравнительно проста в использовании и помогает овладеть основными навыками, необходимыми для работы в других системах управления пересмотрами версий. Во-вторых, *RCS* удобна для резервного копирования файлов конфигурации и для ведения небольшой истории откатов (undo/redo). Если вам не понравятся сделанные изменения, очень легко вернуться к более ранней версии.

RCS возвращает нас во времена младенчества совместной разработки ПО, и как таковая предлагает скудный набор функций по сравнению с более современными конкурентами. Крупнейший недостаток заключается в том, что *RCS* умеет работать только с отдельным файлом. Вашей первой мыслью станет: «Ну и что с нее толку?». И верно, из-за своих ограничений *RCS* – плохой выбор для управления проектом, но случается, что и с отдельным файлом надо поработать, и узконаправленный подход *RCS* будет более оправдан.

Другое ограничение – в одно и то же время с файлом может работать лишь один разработчик. Один файл, один активный редактор: это не делает *RCS* самой гибкой системой в мире. Если ваши требования к редактированию отдельного файла выходят за пределы элементарного сотрудничества, ищите дальше.

Будучи старой, и официально входящей в состав проекта GNU, *RCS* распространена повсеместно – вы найдете ее предустановленной на многих системах, а если нет – инсталляция обычно сводится к простому щелчку мышью в менеджере пакетов вашего дистрибутива. Сейчас это скорее исторический экспонат, чем действительно полезный инструмент (мы бы использовали ее для случайного конфигурационного файла, но вряд ли для чего-то еще), но она служит хорошим введением в мир систем управления версиями, и вряд ли отпугнет слишком многих людей.

«Неплохое введение в системы контроля версий.»



› Графический интерфейс, пожалуй, чрезмерно сложен для простой командной строки *RCS*.

LINUX FORMAT Вердикт

RCS

Версия: 5.7
Сайт: www.gnu.org/software/rcs/rcs.html
Лицензия: GPL

» *Малость староват, чтобы рассматриваться всерьез, но раз уж он установлен в большинстве систем по умолчанию, его, возможно, стоит попробовать.*

Рейтинг 3/10

CVS

Второе открытие RCS для тысяч хакеров.

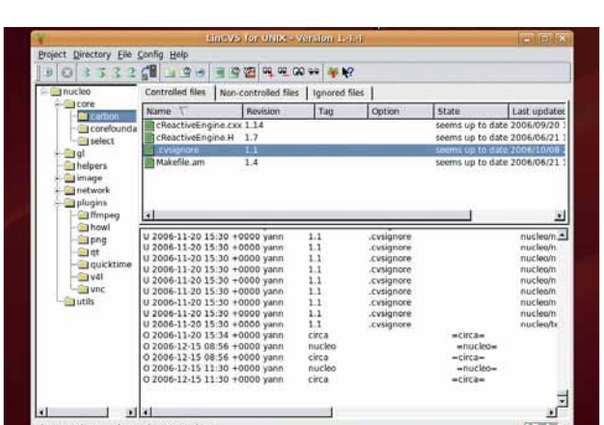
Именно с *CVS* все становится серьезным. Она была разработана для решения основной проблемы *RCS*: ограниченности работой с отдельным файлом. Дик Груне [Dick Grune], первоначальный разработчик *CVS*, нуждался в инструменте для совместной работы над проектом компилятора C с двумя своими студентами, поскольку расписания всех троих резко отличались.

CVS отпочковалась от ранней версии *RCS* и по-прежнему использует тот же формат файлов для хранения истории каждого отдельного файла. Но запуск команд *RCS* применительно к *CVS*-репозиторию может вызвать ряд серьезных проблем, потому что *CVS* управляет не одним файлом, а целым деревом.

CVS также решает проблему множества пользователей за счет использования клиент-серверной модели. Наличие разработчиков, параллельно работающих над одними и теми же файлами (отсюда и полное имя системы – Concurrent Version System, система управления параллельными версиями) означает, что для обработки изменений *CVS* нуждается в сервере. Она блокирует файлы, которые изменяются, или предоставляет простой доступ только на чтение. Сервер *CVS* также способен управлять более чем одним проектом (они хранятся в виде «модулей»). Хотя команды обработки файлов во многом похожи (checkout, update, commit), их теперь нужно исполнять из оболочки *CVS*, а не из командной строки, как это было в *RCS*.

Также усложнился запуск или подключение к серверу, но это не слишком обременительно для среднего программиста. *CVS* по-прежнему хороший выбор, и, подобно *RCS*, устанавливается по умолчанию с большинством операционных систем Linux, получая при этом преимущества повсеместной поддержки. Многие инструменты и приложения Linux все еще собираются с использованием *CVS* – это по-прежнему самая популярная система управления версиями – и практически для каждой крупной ОС существует масса графических клиентов на выбор.

«Лидирует по части поддержки в Linux.»



› *CVS* используется тысячами отдельных проектов.

LINUX FORMAT Вердикт

CVS

Версия: 1:1.12
Сайт: www.nongnu.org/cvs
Лицензия: GPL

» *CVS является хорошим выбором, просто потому, что многие другие разработчики уже используют ее. Но вы ее перерастете.*

Рейтинг 6/10

Aegis

Первая альтернатива клиент-серверной модели.

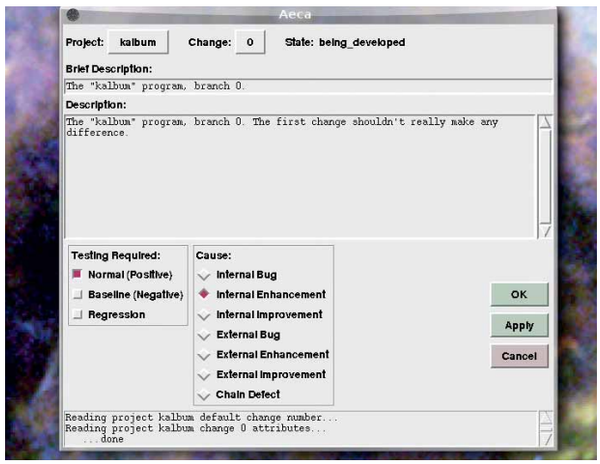
Впервые выпущенная в 1991 году, *Aegis* применяет иной подход к управлению версиями, заметно отличаясь от *Subversion* и *CVS*. Например, изменения децентрализованы, а цикл разработки управляется тестами. Последнее – это методология экстремального программирования, где программист обязан предъявить несколько тестов, прежде чем включить в приложение новую функцию. В результате ваша локальная «песочница» включает только те файлы, с которыми работаете вы, что приводит к путанице, если вы привыкли иметь полную рабочую копию всего проекта.

Это – часть крупнейшей проблемы *Aegis*: трудности использования. Хороший пример – настройка репозитория. Прежде чем приступить к использованию *Aegis*, вам нужно создать новый проект, содержащий несколько файлов конфигурации. Для этого проекта желательно создать отдельного пользователя, отчасти потому, что модель безопасности *Aegis* полностью полагается на права доступа к файлам, а отчасти потому, что это обеспечивает приличный уровень безопасности. Для предоставления доступа другим пользователям используются права группы, а значения umask определяют доступ группы к каждому конкретному проекту.

Aegis успешно использует функции Unix, реализованные в любой Linux-системе. Здесь нет интегрированной поддержки совместной работы по сети, но проект можно легко разделять, используя стандартные протоколы типа FTP, HTTP и NFS. Документация даже хвастает поддержкой «беговой сети», когда участники берут ноги в руки и мчатся по коридору к компьютеру, куда нужно скопировать проект. Не слишком удобно для больших расстояний.

Эта сложность отчасти компенсируется документацией хорошего качества, доступной в сети, хотя и в виде плохо отформатированного HTML. Есть несколько графических интерфейсов (использующих *Tk*) для общих команд, и отличная web-оболочка, но они не конкуренты многим сторонним приложениям, доступным для *CVS* или *Subversion*.

> «Документация отчасти компенсирует ее сложность.»



› Даже имеющиеся графические интерфейсы показывают, как сложна и трудна в понимании *Aegis*.

LINUX FORMAT Вердикт

Aegis

Версия: 4.22
Сайт: http://aegis.sourceforge.net
Лицензия: GPL

» *Неудобна и сложна в использовании, но ее особый набор функций может оказаться созвучным вашему стилю разработки.*

Рейтинг 5/10

Monotone

Система с собственным сетевым протоколом.

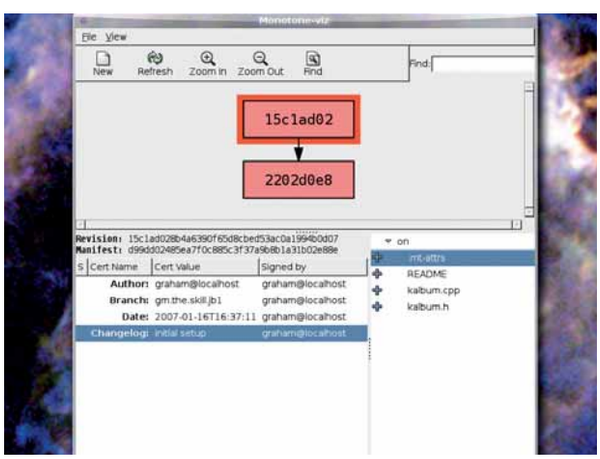
Monotone – это управление версиями для нынешнего поколения разработчиков. Здесь нет старых технологий или старого исходного кода. Выбран распределенный подход, без центрального сервера. Вместо этого каждый клиент отвечает за синхронизацию изменений со всеми другими – во многом тем же способом, каким работает протокол предоставления файлов в общий доступ. Он работает за счет сохранения локальной копии каждого изменения в базе данных SQLite и сравнения версий файлов с использованием алгоритма хэширования SHA1.

Установка проста, и полная версия превосходит *Aegis*. Используя командную строку и команду *Monotone*, вы сперва создаете базу данных, затем генерируете пару ключей SHA1 для подписывания своих файлов. Пользователям *Subversion* и *CVS* покажется очень знакомой работа репозиторием. Команды add (добавление), status (состояние) и commit (фиксация) имеют синтаксис, почти идентичный *CVS*.

Различие проявляется при слиянии ваших изменений с ветвями разработки других людей. Вам сначала нужно экспортировать свой открытый ключ и получить копию от ваших коллег. Затем они должны импортировать ваш общий ключ в «связку ключей» своего *Monotone*. После добавления имен коллег в локальный файл разрешений круг доверия замыкается, и вам нужно просто запустить команду server в *Monotone*. После этого каждый зарегистрированный пользователь сможет синхронизироваться с вашей рабочей копией дерева разработки, используя команду sync. Для копирования данных через интернет или локальную сеть *Monotone* использует собственный протокол, NetSync. Но поскольку он использует один порт (4691), его довольно легко пробросить через SSH-туннель – чтобы обойти слишком усердные брандмауэры.

Единственный недостаток – отсутствие графических инструментов, помогающих управлять репозиторием. Единственный существующий вариант, *Monotone*-viz, великолепен для отображения ветвей проекта, но слабо подходит для чего-то еще.

> «Документация отчасти компенсирует ее сложность.»



› Графических интерфейсов, помогающих работать с *Monotone*, нету, но вот этот украсит плоды ваших трудов.

LINUX FORMAT Вердикт

Monotone

Версия: 0.32
Сайт: www.venge.net/monotone
Лицензия: GPL

» *Monotone сочетает свежий подход к управлению версиями с прекрасной документацией, эффективностью и энтузиазмом.*

Рейтинг 7/10

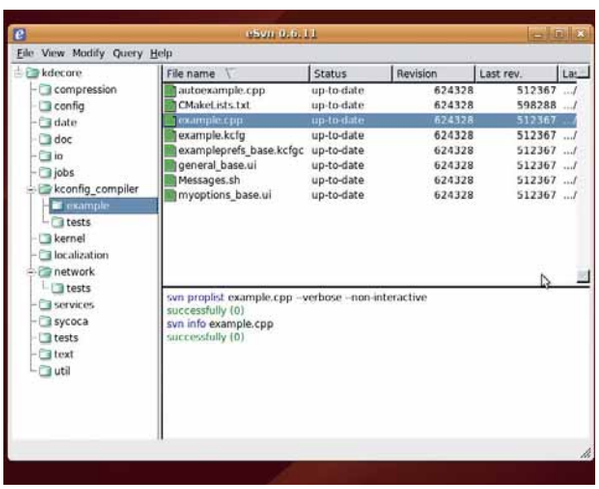
Subversion

Набирающая популярность альтернатива CVS, богатая графическими оболочками.

Когда вы какое-то время используете *CVS*, определенные моменты становятся настоящей проблемой. Первый – любые файлы и каталоги, которые вы перемещаете в пределах своей локальной копии, не будут учтены в изменениях. Они ускользают от радара *CVS*, и их нужно добавлять как новые файлы; а значит, теряется вся накопленная история и информация об изменениях файла или каталога. Поначалу это не проблема, но когда вы в проекте шесть или двенадцать месяцев, это превращается в настоящую пытку.

Другая серьезная беда *CVS* заключается в том, что 'commit' не выполняется одним махом, атомарно. Вместо этого он запускается для каждого изменения, насколько это ему удается. Если разработчику случается в это время редактировать файл, значит, ему не повезло. *Subversion* решает обе эти проблемы, а также исправляет другие недочеты *CVS*, например, обеспечивает версионность символических ссылок и реализует поддержку двоичных файлов.

В результате *Subversion* довольно быстро вытесняет *CVS* как предпочтительную систему управления версиями, и многие выдающиеся проекты, включая *GCC*, *Samba*, Mono, *Apache*, Python и KDE, переключаются на него для управления исходным кодом.



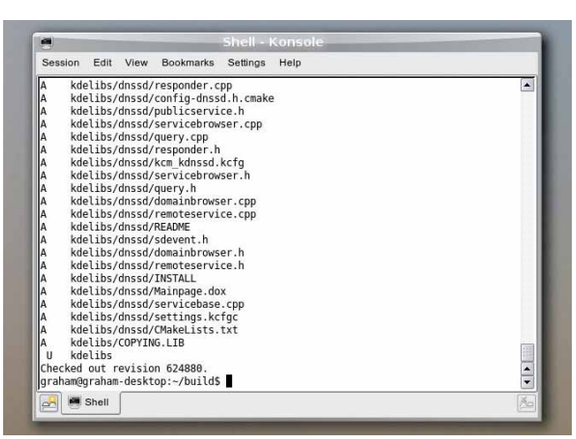
› У *Subversion* больше графических инструментов, чем у любой другой системы управления версиями.

Без боли

Нет сомнений, что *Subversion* популярна именно благодаря сходству с *CVS*. Она использует ту же самую клиент-серверную модель, и основная масса ее команд – двойники команд *CVS*. Кроме переноса истории версий вашего проекта, переключение с одной системы на другую проходит относительно безболезненно. Зачастую вам может сойти с рук простая замена команды *CVS* на svn-эквивалент. Установка прямолинейна, и вы найдете пакеты *Subversion* почти в каждом дистрибутиве Linux, вышедшем за последнюю пару лет.

Клиент полезен и сам по себе, а в некоторых случаях он может оказаться единственным способом скачать версии разработчика какого-нибудь интересного вам проекта. Вы также обнаружите изобилие графических интерфейсов, пытающихся сроднить командную строку с преимуществами графических инструментов. Популярный выбор – *ksvn* для KDE и более общий *esvn*. Поддержка *Subversion* также встроена во многие интегрированные среды разработки, включая *KDevelop*, *Eclipse* (с расширением *Subclipse*), *Zend Studio* и *Xcode* от Apple.

Что касается доступа к серверу, большинство крупных проектов применяют в этом качестве популярный модуль *Apache* **WebDAV**, предоставляющий доступ к web-серверу на чтение-запись. Установить его не сложно, и раз уж модуль *Apache* загружен должным образом, настройка тоже дело простое. Вы, скорее всего, обнаружите, что ваш дистрибутив включает этот модуль, заранее скомпилированный для вашей версии *Apache*. Есть и более простое решение, удобное для небольших групп разработчиков – использовать интегрированный в *Subversion* протокол SVN, как сам по себе в доверенной сети, так и через SSH-туннель, если ваши коллеги разбросаны по всему Интернету.



› Управление из командной строки – по-прежнему единственный реальный способ получить преимущества от всех функций *Subversion*, особенно удаленно.

Subversion – не самое эффективное ПО для управления версиями. Полная копия репозитория сохраняется в скрытом каталоге, что имеет свое преимущество – можно вносить изменения, когда сеть недоступна – но требует очень много места. База данных истории изменений также может расти экспоненциально, по мере внесения изменений всеми разработчиками, и вам необходимо обеспечить регулярное резервирование базы данных Berkeley. Сценарии резервирования предусмотрены, и они присоединяются к «ловушкам» (hooks). «Ловушки» вызываются определенными событиями, например, внесением изменений (commit) разработчиком, и предоставляют эффективный способ настройки сервера *Subversion* под ваши нужды.

Благодаря своей повсеместности, *Subversion* обещает более высокий уровень поддержки сообществом, чем многие другие системы управления пересмотрами. Есть книги, форумы, списки рассылки и свободно публикуемая документация; они помогут вам приступить к работе. Многие проблемы можно решить, скопировав сообщение об ошибке в строку поиска Google, чего нельзя сказать о *Monotone* или *Bazaar*. Добавим сюда относительную простоту установки репозитория *Subversion* или присоединения к открытой разработке, размещенной на другом сервере (как SourceForge, так и Google Code сейчас используют *Subversion*), и *Subversion* становится непревзойденной.

«Обещает более высокий уровень поддержки, чем многие другие системы.»

LINUX FORMAT Вердикт

Subversion

Версия: 1.4
Сайт: http://subversion.tigris.org
Лицензия: Apache/BSD

» *Subversion – вне всякой конкуренции. Она комплексна, предельно модифицируема и отлично поддерживается.*

Рейтинг 9/10

Git

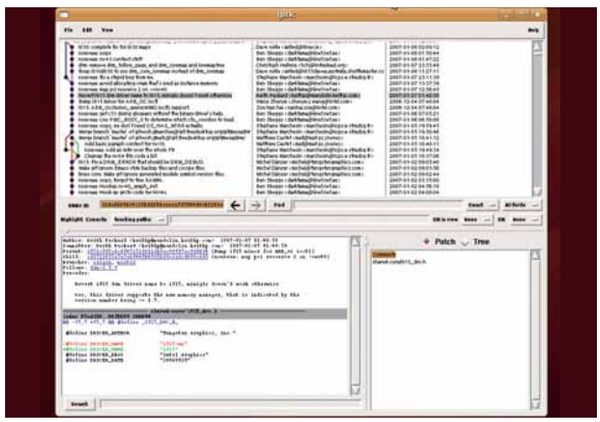
Кто сказал, что Линус был сердит?

Даже если вы не интересуетесь программами управления ревизиями, имя *Git* вам, вероятно, попадалось на том мощном форуме, Slashdot. *Git* – это еще один плод разума Линуса Торвальса, и появился он со скандалом. В мире «До *Git*» многие основные разработчики ядра Linux использовали закрытый инструмент управления пересмотрами под названием *BitKeeper*. Но когда в начале 2005 г. было объявлено об отзыве бесплатной версии *BitKeeper*, в ответ на поднявшийся в сообществе шум Торвальдс разработал *Git*. Он объявил, что *Git* – это вам не просто система управления ревизиями, а скорее усовершенствованная файловая система. Прежде всего, однако, *Git* разрабатывался с прицелом на производительность. Когда вы работаете с проектом масштаба ядра Linux, производительность – это все.

Подобно *Monotone*, *Git* – распределенная система, без сервера. Но в то же время, в Интернете есть серверы, используемые в качестве центрального репозитория для определенных проектов: прекрасный пример – **www.kernel.org**. Различие между использованием их и чего-то вроде *Subversion* заключается в том, что если данные на **kernel.org** будут утрачены, последнюю версию ядра можно будет снова собрать из кода, распределенного среди разработчиков, потому что каждый из них имеет собственную рабочую копию репозитория в подкаталоге **.git**.

Минуло почти два года, и *Git* настолько упрочил свои позиции, что вы найдете его либо установленным по умолчанию, либо на расстоянии всего нескольких щелчков мышью в менеджере пакетов каждого свежего дистрибутива Linux. Несмотря на свою репутацию инструмента для хакеров, он сравнительно прост в использовании. Например, репликация удаленного репозитория – это обычно трудоемкая работа. Но в случае с *Git* это легко. Просто выполните **git clone**, указав URL удаленного репозитория. Другие команды работают подобно аналогичным функциям в *Subversion*, и есть прекрасная утилита для обеспечения взаимодействия между двумя системами.

> «Каждый разработчик имеет копию репозитория.»



› *Gitk*, графический интерфейс *Git*, превращает уродство управления версиями в поэзию. Включен в стандартную поставку *Git*.

LINUX FORMAT Вердикт

Git

Версия: 1.4
Сайт: http://git.or.cz
Лицензия: GPL

» Всего двух лет от роду, Git является высокопроизводительным, серьезным решением для крупнейших проектов.

Рейтинг 7/10

Bazaar

Система управления версиями, спонсируемая Canonical.

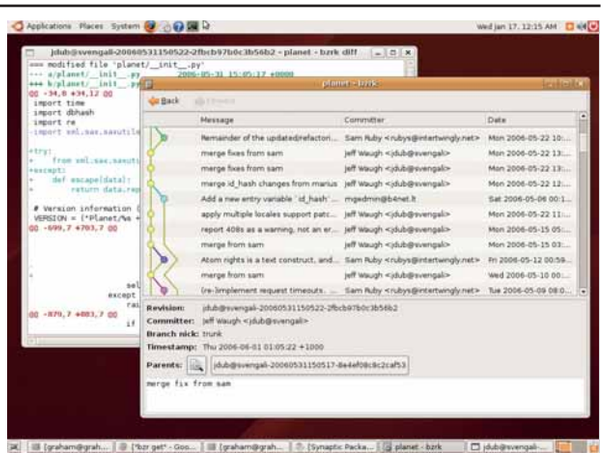
Bazaar – наследник распределенной системы управления версиями, *Bazaar-NG*, возникший благодаря спонсорской поддержке от Canonical Ltd. Неудивительно, что *Bazaar* используется разработчиками Ubuntu и старается быть как можно более понятным. Хотя он использует распределенную модель, многие из команд очень похожи на команды *CVS* или *Subversion*. Поскольку центральный сервер отсутствует, вы заметите различия только когда захотите предоставить в общий доступ свою локальную рабочую копию – вам нужно будет вытянуть (*pull*) изменения с удаленного репозитория и объединить (*merge*) со своими изменениями.

Документация великолепна, включая учебники, охватывающие сложные темы, например, отслеживание главной версии и интеграцию с *CVS*. Базовый учебник даже проводит вас через ряд основных концепций, характерных для всех утилит управления версиями – прекрасный букварь для начинающих. Публикация вашей работы также освежающе проста. Просто скопируйте содержимое своего рабочего каталога на web-сервер, и другие разработчики смогут забирать копии вашей ветви оттуда. Вы можете синхронизировать ее со своей рабочей копией, используя *rsync*; в протокол также встроена поддержка SFTP, вызываемая через команду *push*.

Bazaar использует интерфейс расширений для обеспечения дополнительных функций. Например, несколько графических компонентов на *GTK2* можно запускать прямо из командной строки *Bazaar*. Скажем, ввод **bzr visualize** откроет визуализатор ветви, который выглядит подобно визуализатору *Git*. Есть также графические компоненты для команд *commit*, *diff*, *annotate* и *branch*.

Bazaar трудно не полюбить. У него тот самый разумный подход, который создал популярность Ubuntu, а графические интерфейсы здорово помогают людям, напуганным множеством опций командной строки.

> «Публикация вашей работы очень проста.»



› Пользовательский интерфейс *Bazaar* – один из лучших среди систем управления версиями; идеален для отслеживания изменений и просмотра различий в версиях.

LINUX FORMAT Вердикт

Bazaar

Версия: 0.11
Сайт: www.bazaar-vcs.org
Лицензия: GPL

» Bazaar имеет великолепные графические инструменты и прост в понимании. Наша любимая распределенная система управления версиями.

Рейтинг 8/10

Системы управления версиями

Вердикт

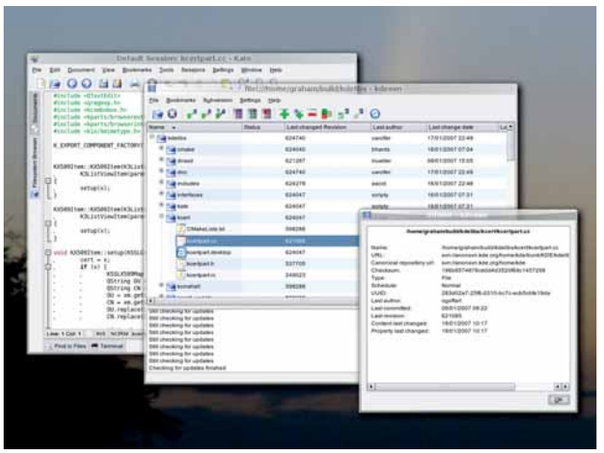
Subversion 9/10

Если честно, мы приступали к этому Сравнению, думая, что это будет игра в одни ворота. Все мы использовали *Subversion*, были знакомы с тем, как она работает, и в большинстве наших проектов также используется *Subversion*. Кому ж еще победить?

Что ж, так и вышло, но не с такой легкостью, как мы думали. Распределенные модели, используемые *Monotone*, *Bazaar* и *Git*, удивительно просты в использовании, а отсутствие центрального сервера – реальная инновация. Если бы мы выбирали распределенное решение, это был бы *Bazaar*. Благодаря разработчикам из Canonical, он выглядел самым ухоженным и проработанным из трех. Но и *Git*, и *Monotone* удивили нас своим удобством, и в следующие двенадцать месяцев мы можем стать свидетелями захвата ими значительной доли рынка за счет некоторых других систем.

В конечном итоге, никто не обогнал *Subversion*. Здесь больше функций, чем в любой другой рассмотренной нами системе, выше стабильность и полнее поддержка. *Subversion* также сравнительно проста для каждого, имеющего опыт работы с *CVS*, да и вообще удобна – и для маленьких проектов с парой программистов, и для огромных, имеющих тысячи разработчиков. Именно комбинация проверенной и заслуживающей доверия технологии с разносторонностью, достаточной для удовлетворения потребностей большинства людей, позволила *Subversion* победить. Наличие ряда графических оболочек, как и интеграция во многие популярные IDE, также помогли сделать *Subversion* хорошим выбором для тех, кто не хотел бы надолго оставаться один на один с командной строкой.

Без сомнения, в целом качество систем управления версиями, рассмотренных в этом Сравнении, невероятно высокое. Даже последние двенадцать месяцев продемонстрировали впечатляющий прогресс. Достаточно взглянуть, каких успехов достиг *Git* за столь короткое время, чтобы понять, как быстро все это развивается, и насколько жизненно необходимыми являются системы управления версиями для будущего Linux. И если по усердию, с которым разрабатываются системы управления версиями, судить о будущем Linux, то оно выглядит прекрасным. LXF



› У *Subversion* львиная доля пользовательских интерфейсов и интеграция в IDE – это прекрасный выбор во всех отношениях.

Терминология управления версиями

Несмотря на значительные различия между отдельными системами управления версиями, они используют общие термины:

» **Репозиторий (repository)**. Здесь размещаются все файлы проекта. Репозиторий может размещаться на удаленном сервере, локально на вашем компьютере, или и там, и там.

» **Рабочая копия (working copy)**. Ваша личная версия репозитория, включающая все сделанные вами изменения и модификации. Также известна как «песочница» (sandbox).

» **Фиксация (commit)**. Загрузка в репозиторий изменений, которые вы сделали локально. Если возникнут конфликты, вам нужно будет разобраться с ними.

» **Ветвь (branch)** или **ответвление (fork)**. Пометка набора изменений как изолированных от основной ветви разработки. Удобно для работы над новыми версиями и выпусками исправлений.

» **Выписка (checkout)**. Получение копии проекта из рабочего репозитория. Впоследствии она становится вашей рабочей копией.

Ваше мнение

Кому нужна помощь в изучении использования систем управления версиями, может взять наш учебник *Subversion* в wiki Linux Format (http://linuxformat.co.uk/wiki). Но системы управления версиями – не только для программистов. Сообщите нам, если придумаете какое-то крутое применение для системы управления версиями. Например, в LXF82 мы перевели на *Subversion* наш домашний каталог. Присылайте свои идеи на letters@linuxformat.ru.

> «В целом, качество систем управления версиями, рассмотренных в Сравнении, невероятно высокое.»

Таблица функций

	Aegis	Bazaar	CVS	Git	Monotone	RCS	Subversion
Лицензия	GPL	GPL	GPL	GPL	GPL	GPL	Apache/BSD
Совместная работа	✓	✓	✓	✓	✓	☒	✓
Модель	P2P	P2P	Server	P2P	P2P	Server	Server
Внутренний протокол	☒	☒	✓	✓	✓	☒	☒
Графический интерфейс	✓	✓	✓	✓	☒	☒	✓
Двоичные файлы	☒	✓	☒	☒	✓	☒	✓
Модифицируемость	☒	☒	✓	✓	☒	☒	✓
Web-интерфейс	✓	✓	✓	✓	☒	☒	✓
Максимальное число пользователей (примерно)	50	500	100	1,000	50	5	500

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux

ЛАДИСЛАВ БОДНАР
основатель, начальник, редактор и сотрудник DistroWatch.com.

Почта ненависти

Debian – удивительный проект. Он содержит более 20000 пакетов, над ним работает более 1000 разработчиков-добровольцев по всему миру, и он полностью свободен во всех смыслах. Тот факт, что спустя 15 лет после создания он все еще силен, доказывает мощь интернет-сотрудничества, без всяких финансовых стимулов для участников.

Но один аспект Debian мне очень не по душе: это культура списков рассылки для разработчиков. Любой читающий их найдет немало веток, позорящих проект. Оскорблениями, насмешками и злобой перенасыщены как технические, так и организационные дискуссии.

Будьте любезны

Тревожит то, что эта «культура» расползается по всей Планете Debian и попадает в другие блоги разработчиков. Dunc Tank – спорная схема временных выплат отдельным разработчикам – лишь одна из наиболее значимых тем, вызывающих трения на Планете Debian. А таких тем очень много.

Debian'цы, остановите, пожалуйста, эту бесконечную публичную перебранку! До нажатия кнопки Send в почтовом клиенте или публикации гневной отповеди в вашем блоге, перечитайте пункт 4 Социального соглашения Debian: «Мы исходим из нужд наших пользователей и сообщества свободного ПО. Мы ставим их интересы на первое место». Если вы продолжаете считать, что ваше высказывание – в интересах ваших пользователей, то отсылайте. А иначе нажмите Delete.

ladislav.bodnar@futurenet.co.uk.

Весна в Париже

Mandriva Linux 2007.1 'Spring' Возврат к старому шестимесячному циклу и новый рабочий стол.

Являясь одним из самых романтичных городов мира, Париж умеет превращать обычных людей в художников, графоманов в поэтов, любовные интриги в вечные союзы сердец. Особенно весной.

Пусть компьютерное ПО не столь гламурно, как большинство занятий во французской столице, оно тоже чует приход весны. Mandriva доказала это серией заявлений о представлении нескольких интересных продуктов, один из которых, Metisse – трехмерный рабочий стол, будет интегрирован в дистрибутив.

› Metisse, трехмерный рабочий стол Mandriva, сулит новые пути повышения производительности: например, эффект отгибающихся окон.

Апрельская версия Mandriva Linux 2007 (названная просто Mandriva Spring – Весенняя) обозначит возврат французского дистрибутивостроителя к шестимесячному циклу разработки. Это в духе большинства его соперников и подчеркивает большую скорость разработки программ в мире Open Source: выпуск одного продукта в год приемлем для бизнес-пользователей, обычно не жаждущих супер-обновлений, но большинство линуксоидов желают, чтобы новейшее и лучшее было включено в их дистрибутив как можно скорее.

Одной из горячих новинок, которая удивила (и разделила) сообщество пользователей Linux в 2006, был трехмерный рабочий стол – в форме Xgl и Beryl с Compiz и AIGLX. Хотя мнения об этой технологии сильно разнятся – одни пользователи клянутся, что она осветила их жизнь, а другие презрительно считают её чистой «визуальщиной» – факт, что дистрибутивы, включившие 3D-функции, привлекают больше пользователей. «Привлекательность – это функция» в прошлом году было мантрой Марка Шаттлворта, а трехмерный рабочий стол вносит здоровую дозу приятной миловидности.

> «Рабочий стол Linux уже не в догоняющих, а в лидерах.»

В грядущем релизе Mandriva тоже заготовила сюрприз – новый французский проект по созданию трехмерного рабочего стола (вообще-то целый инструментарий) под названием Metisse. Уже включённый в Live-дистрибутив Mandriva One 2007, Metisse предлагает по-настоящему полезные функции. Например, при выделении текста в окне, частично закрытом другим, верхнее окно временно отгибается, и выделяемый текст не загораживается. Вместо куба, реализованного в Compiz и Beryl, Metisse предоставляет пейджер с четко проработанным рабочим пространством. Магнитные границы, вращение окон, масштабирование, лупа и другие эффекты доступны тому, кто потрудится изучить мириады клавиатурных комбинаций и жестов мышью.

Это, пожалуй, одно из неудобств Metisse: как любое другое сложное ПО, его освоит только тот, у кого есть время продираться сквозь его лабиринты. Разумеется, скептики будут воротить от него нос, как многие фанаты командной строки отмахивались от графического интерфейса в 1980-х. Хотя Metisse не столь революционен, каким был GUI, он приносит много освежающе-новых идей на наши компьютеры, вновь доказывая, что рабочий стол Linux уже не догоняет проприетарные операционные системы, а опережает их.

www.mandriva.com/projects/metisse

Metisse рассматривается на стр. 17 и имеется на нашем DVD.

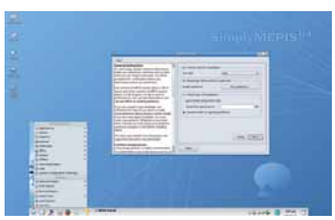
Прибыл Mepis-64

SimplyMepis 6.0-4 Новое лицо и AMD64-версия дружелюбного к пользователю Live-дистрибутива.

Из отзывов пользователей. «Это первая 64-битная ОС, в которой все работает прямо из коробки, – написал восхищенный пользователь на форуме DistroWatch.com, попробовавший SimplyMepis-64 6.0-4. – Это очень важно для меня, среднего пользователя, который просто хочет, чтобы все сразу заработало».

Это действительно комплимент, если учесть, что до этого Mepis не выпускал своей 64-битной версии. В ответ на просьбы AMD64-пользователей, разработчики проекта наконец собрали 64-битную ОС, чтобы те наслаждались стандартными расширениями для браузеров и просматривали многие популярные медиа-форматы без отслеживания и ручной установки отсутствующих частей.

Наше первое впечатление от нового продукта очень выиграло от давно напрашивавшейся ревизии рабочего стола SimplyMepis. Обновились изображение при загрузке, окно входа, оформление окон, набор иконок и обои – рабочий стол заметно похорошел. Хотя базовая система нового



› LiveCD SimplyMepis 6.0-4 включает установщик на жесткий диск.

дистрибутива SimplyMepis с предыдущего издания осталась нетронутой, несколько полезных обновлений пакетов, например, *Firefox 2.0*, усиливают ощущение, что данный релиз заслужил номер 6.1, а не 6.0-4.

Впечатляет также интеграция в SimplyMepis-64 множества проприетарных медиа-расширений: пользователи смогут смотреть сетевое видео и слушать потоковое аудио столь же легко, как в Windows. Хотя пуристы вряд ли придут от этого в восторг, пользователи, предпочитающие удобство идеологии, найдут SimplyMepis практически идеальной заменой своей старой ОС.

www.mepis.org

Свободу демонам!

FreeSBIE 2.0 Редкая Live-ОС на основе BSD достигла магической цифры 2.0.

Linux LiveCD полно на любой вкус и цвет, но разработка Live-версий операционных систем семейства BSD явно засиделась на старте. Их не было до февраля 2004, когда на сцену вышел самый первый LiveCD на основе FreeBSD – FreeSBIE 1.0. Маттео Риондато [Matteo Riondato] со товарищи из итальянской группы пользователей FreeBSD выпустили второй крупный релиз проекта в январе 2007.

FreeSBIE 2.0 – LiveCD, ориентированный на применение в настольных компьютерах и восстановление системы. Он использует Xfce, сравнительно легковесное окружение рабочего стола, и содержит небольшой набор популярных приложений для Интернета, работы с графикой, воспроизведения мультимедиа и офиса. Инструменты управления файлами представлены программами *Xffm* и *Rox-Filer*.

При всей полезности FreeSBIE, ясно, что LiveCD, основанный на BSD, по уровню технологий не сравним с Linux-аналогами. К примеру, FreeSBIE 2.0 вообще не использует новомодные оверлейные файловые системы, разрешающие виртуальную запись данных в режиме LiveCD. Другие функции, обычно предоставляемые большинством Linux LiveCD, вроде загрузки всего CD в память или установку CD на жесткий диск, в FreeSBIE 2.0 также отсутствуют.

› Маттео Риондато, ответственный за выпуск FreeSBIE LiveCD.

Тем не менее, пользователи, знакомые с BSD, сочтут FreeSBIE 2.0 бесценным продуктом – особенно те, кому необходимо выполнять действия по восстановлению или демонстрации FreeBSD новым пользователям. LXF

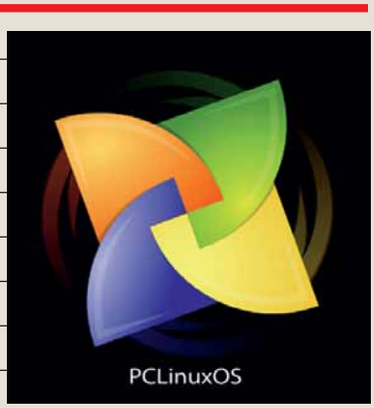
www.freesbie.org

Как быстро загружается ваш дистрибутив?

За последние месяцы в сообществах различных дистрибутивов мы наслушались разговоров об альтернативах старому методу загрузки Unix и других улучшениях, призванных заставить компьютеры стартовать быстрее. Ввиду того, что время загрузки зависит от множества независимых факторов (новизны оборудования, количества запускаемых служб, проверки стартовым скриптом наличия нового оборудования и т.д.), разработчики разных дистрибутивов реализуют различные идеи, ускоряющие процесс загрузки.

Мы протестировали время загрузки нескольких популярных дистрибутивов в их стандартной настольной конфигурации на машине с процессором Pentium 4 1,4 ГГц с 384 МБ ОЗУ. Результаты сведены в таблицу внизу. В столбце «Время загрузки» приведено количество секунд, прошедшее с появления приглашения загрузки до появления менеджера входа KDE или Gnome.

Дистрибутив	Время загрузки (сек.)
PCLinuxOS 2007	33
Mandriva Linux 2007	40
Debian GNU/Linux 4.0	47
Slackware Linux 11.0	48
Ubuntu 6.10	49
SimplyMepis 6.0-4	51
OpenSUSE 10.2	61
Fedora Core 6	72
Xandros Desktop 4.1	89



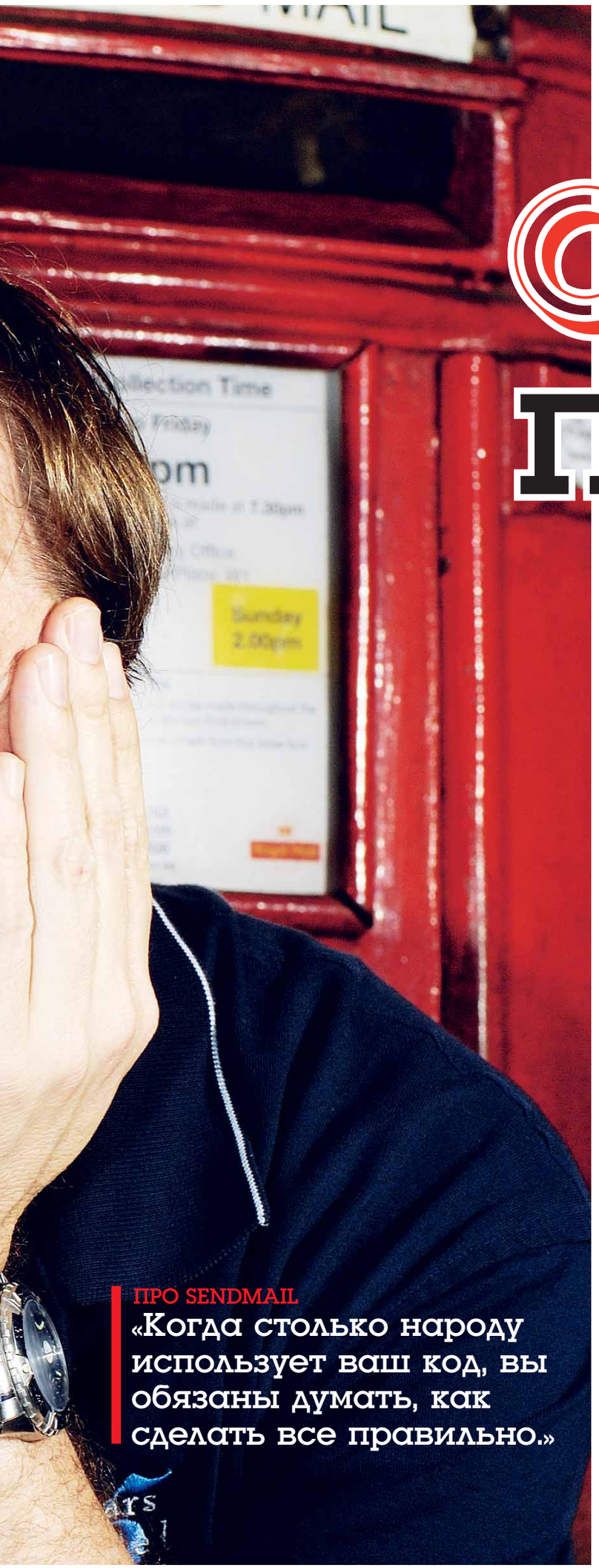
Хит-парад дистрибутивов

10 самых посещаемых страниц на DistroWatch.com за период с 02/01/07 по 01/02/07 (среднее число визитов в день)

Дистрибутив	Число визитов	
1 Ubuntu	2,588	↑
2 SUSE	2,059	↓
3 PCLinuxOS	1,575	↑
4 Fedora	1,358	↓
5 SimplyMepis	1,241	↓
6 Debian GNU/Linux	1,038	↓
7 Sabayon Linux	1,001	↓
8 Mandriva	995	↔
9 Damn Small Linux	841	↑
10 Knoppix	795	↓

» DistroWatch.com отслеживает популярность дистрибутивов, основываясь на количестве посещений сайтов, посвященных конкретным дистрибутивам. Хотя эти цифры и не отражают реальное количество инсталляций, они являются индикатором популярности дистрибутива на данный момент времени.

Photography: Rob Scott



ПРО SENDMAIL
«Когда столько народу использует ваш код, вы обязаны думать, как сделать все правильно.»

Отец почты

В этом месяце мы беседуем с **Эриком Оллменом**, одним из отцов-основателей электронной почты, о былых и насущных проблемах системы связи, которая в наши дни рассматривается многими как основное гражданское право…

Будучи в начале семидесятых студентом в Беркли, Эрик Оллмен [Eric Allman] работал с кодом Unix вплоть до победного итога – Mail Transport Agent, охватывающего ARPANET. В 1981 код MTA развился в *Sendmail*, до сих пор являющийся самым широко используемым почтовым сервером в мире. В 1998 году Оллмен основал Sendmail Inc, чтобы финансировать разработку и увеличить ее полезность для мира бизнеса. Он по-прежнему живет в Калифорнии, но недавно съездил в Великобританию, где обменялся сигналами с **Ником Вейчем**.

Linux Format: Каковы основные проблемы e-mail сегодня?

Эрик Оллмен (ЭО): Если отложить в сторону очевидные проблемы – спам, фишинг, вирусы – и рассмотреть проблемы более тонкие, то я думаю, в каком-то смысле e-mail используется недостаточно и в то же время чрезмерно. Множество людей используют его зря – а в этом случае e-mail становится настоящим пожирателем времени. Приведу пример: два человека активно обмениваются сообщениями, обсуждая некое проектное решение или что-либо подобное, а зачастую пять минут телефонного разговора сэкономило бы им полдня. Вот это и есть то самое «чрезмерное использование».

Одна из приятных вещей в e-mail – или неприятных, в зависимости от ситуации – заключается в том, что e-mail оставляет документальный след. Можно, например, собрать все предложенные решения, а затем один человек сообщит всем по электронной почте, какой он видит финальную резолюцию. Попробуйте собрать в одной комнате 15 человек для обсуждения проектного решения – можете дать гарантию, что ничего из этого не выйдет.

На самом деле ничего нового в этом нет – это было актуально для всех средств связи во все времена, насколько я могу судить – но e-mail определенно страдает от этого. И, возможно, даже актуальнее то, что сегодня e-mail содержит все больше и больше интеллектуальной собственности компании, и поэтому компаниям приходится обращать на нее все больше и больше внимания. E-mail может стать причиной очень существенной утечки информации, не потому, что кто-то вредит намеренно, а просто потому, что люди, бывает, ошибаются адресом или просто не задумываются – отправляют кому-то сообщение, не указывая, что содержащаяся в нем информация…

LXF: Конечно – по-моему, в книге Кевина Митника [Kevin Mitnick] по социальной инженерии приводятся примеры, как легко добыть действительно секретную »

» **информацию через электронную почту, потому что люди, видя e-mail-сообщение, верят, что оно пришло от того, кому можно доверять.**

ЭО: Фишинг похож на то, как фокусники работают руками: они фокусируют [вот и происхождение слова выясняется! – прим. ред.] ваше внимание «там», чтобы вы не смогли увидеть, что они делают «здесь». И некоторые фишеры весьма и весьма искусны.

LXF: По последним данным, не менее 70% всего e-mail-трафика занимает спам; потом идут фишинговые атаки и все остальное, связанное с e-mail. Не приближаемся ли мы к тому, что проблемы перекроют всю пользу? Что, система полностью разрушена?

ЭО: Ну, ответов у меня два. Один: «Проблема действительно есть», и я считаю, что спам, вирусы, фишинг и т.п. несомненно понизили ценность e-mail. Но вправду ли все так безнадежно? Я так не думаю. Спам – одна из причин, по которым я работаю над штукой под названием DKIM, почтой, идентифицированной доменными ключами (domain keys identified mail). Это способ авторизации e-mail, и одна из причин появления данной технологии следующая: можно многое сделать, если известно, от кого в действительности пришло сообщение. Сейчас все, на что мы способны – это сканировать содержание письма: «посмотрим, о чем оно, и попробуем угадать, спам оно или нет». Проблема в том, что пока вы можете только принять это. Вы можете сделать многое – иногда спам настолько явно является спамом, что вы можете сразу избавиться от него.

DKIM способствует своего рода фильтрации на базе идентификации. Сочетание все улучшающихся фильтров контента с идентификационной фильтрацией, я думаю, приведет нас к желаемому, [но спам как проблема] никогда не исчезнет. В мой настоящий почтовый ящик тоже попадают рекламные листовки, никуда не денешься, но, по крайней мере, перед моей дверью не останавливается грузовик и не вываливает на меня кучу рекламных материалов – а в e-mail как раз происходит нечто подобное.

LXF: Вы считаете, бремя решения проблемы спама ложится исключительно на получателей, или провайдеры и другие организации тоже что-то предпримут?

ЭО: Честно говоря, фильтровать спам – это неплохой бизнес.. И [у ISP] имеется свой интерес в фильтрации спама – ведь если они переложат данную задачу на плечи потребителей, то им придется хранить весь этот спам самим. В идеале от спама следует избавляться как можно скорее.

Я думаю, что с предприятиями дело обстоит несколько иначе. Если хотите, большинство пользователей e-mail принадлежат ISP или ESP – Yahoo, Gmails, Hotmail во всем мире – а деньги все-таки у предприятий, потому что для них очень важно время с точки зрения бизнеса. И для них схема немного другая: их работники ведь не уволятся из-за завалов спама в электронной почте, но, с другой стороны, все те же работники будут тратить на спам свое время, так что в фильтрации они тоже заинтересованы. В каком-то смысле лучше бы передвинуть это бремя возможно ближе к центру. Но это сложно. Велось много дискуссий об использовании e-postage и тому подобного, но есть и политические, и технические причины, по которым в ближайшем будущем этого не произойдет.

Есть много возможностей снизить затраты. Один из примеров – DKIM. DKIM, понятно, разработан таким, как есть, не по этой причине, но поскольку необходимо рассчитывать криптографическую функцию от сообщения, отправителю потребуются большие компьютерные мощности.

LXF: Вы думаете, что DKIM будет принят повсеместно?

ЭО: Возьмите DomainKeys, его предшественника. Стандарт еще даже не стабилизировался, а Google уже подписывает все свои сообщения с помощью DK, Yahoo подписывает все свои сообщения с помощью DK, eBay подписывает все свои сообщения с помощью DK. Это – огромный бизнес. По-моему, я слышал, что Amazon тоже собирается использовать его, и я знаю, что многие финансовые учреждения тоже продемонстрировали свой интерес.

LXF: Значит, доведись вам заново создавать инфраструктуру MTA, она, видимо, в корне отличалась бы от изначальной. Если бы вы забыли о том, что электронная почта вообще существовала, и начали сначала, что бы вы реализовали в e-mail 2.0?

ЭО: Если бы я создавал ее сегодня по новой, я бы включил авторизацию в качестве исходного пункта. Надо не достраивать ее потом, а просто создать в самом начале. А в самом начале этого не произошло, потому что – люди этого не осознают – алгоритм RSA, используемый в качестве основы шифрования с публичным ключом, даже не был еще изобретен, когда разрабатывался SMTP. Сложновато требовать того, что пока не изобретено! А теперь говорят: «В e-mail есть ошибки!» Естественно, сейчас мы сделали бы все по-другому, но пришлось бы ждать e-mail лишних лет 15–20.

ПРО TCP/IP

«Линуксоиды создали свой протокол, Бог знает, зачем. Может, в те выходные просто шел дождь!»

Некоторые вещи в протоколе я бы сделал иначе. Они были бы эффективнее: сейчас в них слишком много времени уходит на запросы туда и обратно – ну, знаете, всякие компьютерные мелочи. И, конечно, я бы постарался прояснить значение некоторых заголовков, например, «ответить» [reply to]. Люди никак не поладят насчет того, что под ним подразумевается…

LXF: Вас не удивляет, что спустя столько лет *Sendmail* продолжает занимать командную позицию в инфраструктуре?

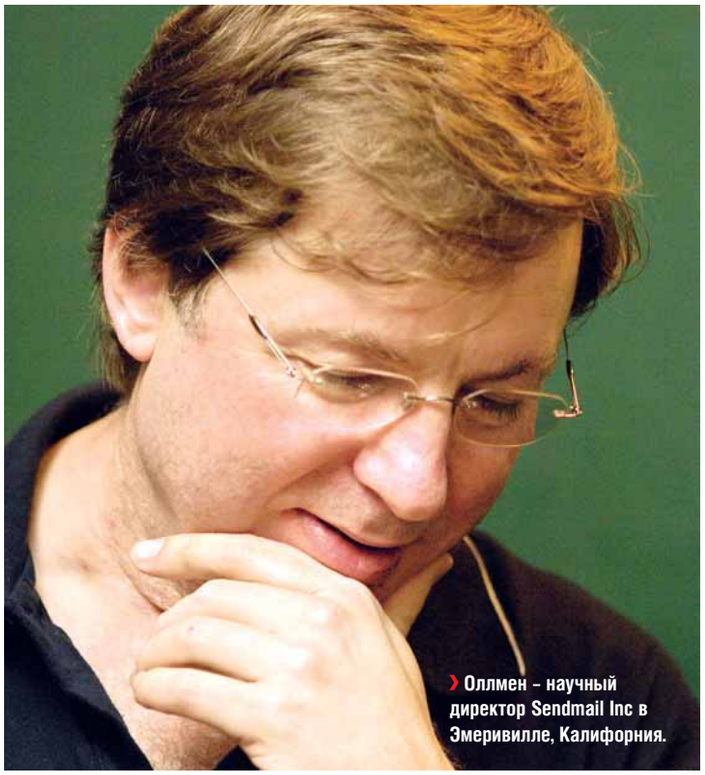
ЭО: В каком-то смысле я удивлен, но не потому, что с *Sendmail* что-то неправильно, а потому, что не так много технологий 25-летней давности, которые все еще работают.

Естественно, что за эти годы *Sendmail* обновлялся, но впечатляет, что он до сих пор на плаву. *Apache* ведь новее… но можно пробежаться по списку программ с открытым кодом, и большинство из них будут старше. Есть стек TCP/IP, ему примерно столько же лет, но он используется во многих системах. Хотя, по-моему, не в Linux – линуксоиды создали собственный, Бог знает, зачем, но создали. Может, в те выходные просто шел дождь!

LXF: Я заметил, что *Sendmail* сейчас часто обновляется. Это потому, что вы всегда наготове с исправлениями по безопасности? По сравнению, скажем, с *Exim*, обновления и заплатки для *Sendmail* выходят в два-три раза чаще.

ЭО: На самом деле у нас в Sendmail Inc было много споров. Мы стали очень заботиться о проблемах безопасности. Мы относим к проблемам безопасности то, что другие таковыми не считают. И мы выпустили заплатки безопасности для *Sendmail* из-за ошибок в OpenSSL. К *Sendmail* они отношения не имеют, но ведь конечному пользователю разницы нет. Есть ошибка в системе безопасности, через порт 25 возможна атака, следовательно, это наша проблема. Я думаю, часто наличествует такое отношение: «Код не наш, значит, и проблема не наша». Возможно, мы заняли тоже в своем роде крайнюю позицию по этому вопросу, но мне хочется сделать для пользователя все возможное.

LXF: Я догадываюсь, что если вы предоставляете заплатки, то понимают ли люди, что обновление гарантируется, или не понимают – это уже их дело.



Оллмен – научный директор Sendmail Inc в Эмеривилле, Калифорния.

ЭО: Верно. Знаете, честно говоря, в прошлом у *Sendmail* были проблемы с безопасностью. Я не собираюсь этого скрывать. Но когда-то давно, вроде бы в версии 8.8, мы прошлись по коду. Мы внесли в него проверку, чтобы перед открытием файла он строго-настрого проверялся на предмет наличия разрешения. Возможно, это самая параноидная программа – даже параноиднее, чем *Apache*, где при неправильной настройке системы кто угодно может разгуливать по всем вашим файлам. И такое бывало, но никто ни в чем не обвинял *Apache*, просто говорили: «Вы же неправильно настроили систему». А если вы неправильно настроили *Sendmail*, во всем обвиняли *Sendmail*. Это же несправедливо!

LXF: Ну, справедливо сказать, что настройка *Sendmail* – это само по себе искусство, не так ли?

ЭО: Настройка *Apache* – тоже искусство. Файл конфигурации *Apache* легче читается, но в нем примерно столько же опций, сколько в *Sendmail*, и многие их них довольно сложно понять. Если вам нужен серьезный web-сайт, то понадобится настоящий гуру *Apache*. А если у вас серьезный почтовый сервер и на нем работает *Sendmail*, вам потребуется гуру *Sendmail*. По мне, это не слишком высокая цена. И, кстати, все это говорится вовсе не в упрек *Apache*. *Apache* – мощная программа, и нужно уметь настраивать ее, чтобы использовать всю ее мощь и гибкость.

LXF: Но все же, в некотором смысле, если бы настроить *Sendmail* было бы проще, то было бы меньше плохо настроенных *Sendmail*-серверов.

ЭО: Думаю, вы совершенно правы. Я и перевел *Sendmail* на M4 как раз для того, чтобы изменить ситуацию. Народ по-прежнему упорствует, редактируя конфигурационный напрямую. А я считаю его двоичным файлом. Я не буду заходить в него и править – ну, если мне понадобится очень быстро что-то проверить [*хлопает по запястью*], я зайду, но вам надо держаться подальше от файла **.mc**. Тогда настройка становится намного проще. Не скажу, что она станет простой, но она будет проще.

LXF: А вы лично чувствуете большую ответственность за всех тех, кто использует *Sendmail*?

ЭО: Помню, у меня как-то раз случилось небольшое озарение, и я сказал: «Я не могу оптимизировать *Sendmail* так, чтобы он был хорош для определенной системы отправитель–получатель. Потому что с другой стороны тоже будет *Sendmail*, и если я оптимизирую его для одной системы, это приведет к деградации другой. Мне придется оптимизировать его для всего Интернета, а это означает, что некоторые решения будут выглядеть неоптимальными с точки зрения конкретного узла». Но вы правы. Есть ответственность за правильность действия. Когда столько народа использует твой код, думать об этом надо.

LXF: GPL3 – это хорошо или плохо?

ЭО: Я не анализировал версию 3 подробно, просто бегло просмотрел. Мне показалось, что это – существенное усовершенствование версии 2. Я сильно сомневался по поводу GPL2, но, возможно, подумаю об использовании GPL3. Навскидку, юридический язык стал очень сжатым. Одна из причин, по которым мне не нравилась GPL, и что мне по-прежнему не нравится в 3, заключается в том, что лицензия все-таки юридический документ, а не политическая полемика – это неподходящее место для полемики. Не то что я не согласен с политикой…

LXF: А вас никогда не беспокоило, что если бы у Open Source была Ахиллесова пята, то это был бы вопрос интеллектуальной собственности, просто потому что это Open Source? Вас не беспокоило, что в *Sendmail* может пробраться код, не достойный существовать?

ЭО: Не думаю, что я ответил бы «да» именно на этот вопрос, но я бы ответил «да» на вопрос очень близкий: что кто-то мог бы – или на самом деле может – получить патент на технологию *Sendmail*, а потом придти и сказать: «Кстати, вы принадлежите нам!» Это – одна из проблем с патентами на ПО. Чтобы получить патент, надо опубликовать то, что вы сделали. А некоторые вещи… Патенты выдаются на такие вещи, которые людям в голову не приходит публиковать – настолько они очевидны, а потом кто-то другой получает патент и возвращается – и тогда вы не то чтобы совсем пропали, но придется раскошелиться на дорогостоящих юристов.

LXF: Да, кажется, именно так это и бывает. Это одна из уязвимостей Open Source – у тех, кто пишет программы, обычно недостаточно толстые кошельки, чтобы нанимать адвокатов.

ЭО: Точно. Именно поэтому я в какой-то степени понимаю точку зрения SUSE или Novell, почему они хотят получить гарантию от Microsoft [имеется в виду сделка 'Microvell']. Если вы – корпорация, то вы не смеете прикрывать своих пользователей, пока не будете уверены, что вас самих не потащат в суд, так что я понимаю позицию Novell. Не уверен, что это хорошая идея, но я их понимаю.

ПРО E-MAIL, ЗАДНИМ ЧИСЛОМ

«Если бы я создавал это сегодня, я бы включил авторизацию в качестве исходного пункта.»

LXF: Вы упомянули DKIM – а какие еще интересные вещи на подходе в *Sendmail*? Я уверен, что вы еще над чем-то работаете.

ЭО: На самом деле есть несколько проектов, над которыми я работаю лично. Знаете, одной из важных причин начать Sendmail Inc была в том, что я слишком много времени уделял поддержке *Sendmail*, а я хотел войти в это дело, и снова работать с кодом, и разрабатывать всякие новшества, и т.д. Оказалось, что организация собственной фирмы –не лучший способ вернуться к программированию. Это была большая ошибка с моей стороны! Но с недавнего времени мне удалось выкроить немного времени и начать работу над некоторыми проектами.

Будет ли это одобрено IETF или нет, вполне очевидно, что Китай собирается начать использовать в сообщениях свои языковые символы. Это уже сейчас можно делать в теле сообщения, но в *Sendmail*, из-за способа внутренней кодировки заголовков, это было не чисто 8-битным – ну, сейчас уже все чисто. Как я уже сказал, это не те вещи, от которых ликуют люди, но с моей точки зрения, очень, очень важно их сделать, причем до того, как это перерастет в кризис.

Но все же большинство настоящих инноваций в e-mail касаются фильтров. Релиз 8.14 вышел с новой версией интерфейса milter, и он более мощный, и в milter больше можно сделать. Я думаю, что это главное. А будут ли инновации в *Sendmail*? С технической стороны – нет, 95% наших коммерческих усилий направлено на milter, и именно так и должно быть.

LXF: И в заключение, я прав, что это вы написали исходную версию *Trek* для Unix?

ЭО: Вы совершенно правы.

LXF: Так ведь он просто вопиет об обновлении!

ЭО: Знаете, мне это даже в голову не приходило!

LXF: А я на это столько часов потратил!

ЭО: И я тоже! Знаете, когда-нибудь я, возможно, вернусь к этому, чисто ради удовольствия, вернусь и гляну на него, но он ведь текстовый, а народу подавай графику, чтобы все сияло и блестело. LXF

Читайте больше!

Узнайте, что Эрик думает о Microvell, патентах и Open Source на www.linuxformat.co.uk/mag/allman.html.

Что за штука... XHTML 2?

Еще один рывок Web-стандартов в сторону XML – **Пол Хадсон** все объяснит…

Да, да, я знаю, что вы хотите сказать: XHTML хорош, потому что он несет в себе смысл, а HTML полон бесполезных стилевых тэгов вроде <font>, поэтому мы все должны пользоваться XHTML.

Да, именно это я хотел сказать. Почти. Повнимательнее взглянув на заголовок этой статьи, вы заметите, что речь идет скорее о XHTML 2, чем о XHTML вообще.

Он что, в два раза педантичнее XHTML 1?

Вообще-то он очень отличается от XHTML 1 и 1.1, потому-то его и ждали так долго. Язык XHTML 1 был стандартизован в 2001 г., в попытке применить некоторые правила XML к HTML. Например, нельзя было использовать

```
<strong><em>Foo</strong></em>
```

потому, что извлекать элементы из стека необходимо в порядке, обратном тому, в котором их туда помещали:

```
<strong><em>Foo</em></strong>
```

было бы верно.

А еще, каждый атрибут XHTML надо было заключать в кавычки, даже если это выглядело глупо, например **selected="selected"** для отмеченного флажка.

Почему же так долго ждали XHTML 2?

По сути, XHTML 1.1 – тот же HTML 4, наряженный в парадный костюм. Любой браузер, способный прочесть HTML 4, мог также читать и XHTML 1.1, ведь они неотличимы.

Идея заключалась в том, что XHTML-страницы могли обслуживаться серверами используя MIME-типы application/xml или application/xhtml+xml, и браузеры могли воспринимать документы как XML. Подразумевалось, что браузер будет игнорировать код обработки скверного HTML, зато на ура примет утонченный XML. К несчастью, этого так и не случилось. Некоторые браузеры – включая *Internet Explorer* версий 6 и 7 – не умеют толком обрабатывать MIME-тип XHTML, а это значит, что XHTML-страницы по-прежнему обслуживаются (и читаются) как HTML.

Вы не ответили на мой вопрос…

Разве? Ах, да. Так вот, XHTML 1 был очень похож на HTML 4. Но W3C нужен был XHTML 2, чтобы исправить многие недостатки HTML, а это означает отсутствие обратной совместимости с HTML. Оказалось, что некоторые части XHTML 2 обратно-несовместимы даже с XHTML 1.1!

Пожалуйста, конкретнее, не то мы и в четыре страницы не уложимся.

Хорошо, а помните, когда вы делали первые шаги в XHTML, валидатор W3C XHTML отказывался признавать вашу страницу, пока ко всем рисункам не добавят атрибуты ALT?

О, да. Это был не мой день…

А вот в XHTML 2, боюсь, вам придется убрать все эти атрибуты, потому что их больше нет в спецификации. Перемена имеет смысл, так как в XHTML 2 любой элемент может содержать как объект (например, картинку), так и замещающий текст. Первичный контент используется как атрибут элемента, а вторичный (который замещает основной, если он не может быть загружен или еще не загрузился) помещается внутрь элемента. Например:

> «Помните, как вы добавляли атрибуты ALT ко всем рисункам? Боюсь, теперь придется их удалять.»

```
<img src="tacheback.gif"><strong>Andrew Gregory</strong> winking like a mad man</img>
```

и

```
<p src="tacheback.gif"><strong>Andrew Gregory</strong> winking like a mad man</p>
```

дадут одинаковый результат. Выгода в том, что теперь вы можете использовать XHTML-форматирование и в альтернативном тексте – например, часть **Andrew Gregory** будет выделена полужирным.

Придется позлобствовать, но преимущества просматриваются.

Хорошо! Пришло время упомянуть о другой долгожданной перемене: тэги <b> и <i> удалены. XHTML 1.0 удалил множество стилевых тэгов HTML, но оставил наиболее популярные, <b> и <i> – на них построено слишком много сайтов. За время, прошедшее со времени стандартизации XHTML 1.0, многие сайты перешли на и , и есть надежда на то, что большинство новых сайтов даже не вспомнит о существовании <b> и <i>.

Не понимаю, за что тут бороться…

За смысловую нагрузку элементов. Например, тэг <i> означает «написать это курсивом (italic)», а означает «эту часть текста необходимо специально выделить (emphasis – выделение, акцент)». Чаще всего браузеры выделяют курсивом, но в конечном счете, как именно выделять данный текст – это дело браузера. Для людей, пользующихся озвучиванием экранного текста, может быть выделено особой интонацией или сказано немного громче – не так-то просто говорить курсивом, поэтому <i> просто не имеет смысла. С тэгом <b> тоже были проблемы, но основные неурядицы связаны с таблицами стилей. Со

 шрифт можно слегка увеличить и окрасить, например, красным, чтобы подчеркнуть его значение, а вот текст с <b> можно и не заметить, если он в действительности не полужирный!

Не особо радикальные перемены. Почему они так долго готовились?

Бесспорно, крупнейшее событие в XHTML 2 – применение XForms как стандарта для форм ввода данных. Формы HTML – необычайно простой способ приема пользовательского ввода, а вот XForms, если честно, нет. Но дополнительной сложности сопутствует и дополнительная мощь: можно заполнять XForms заранее с помощью XML, отправлять их данные с помощью XML и даже отправлять и получать данные без обновления страницы – почти как в AJAX. Для изучения нового синтаксиса понадобится целый учебник – если буквы M, V и C кое-что значат для вас, вы должны понять, как соотносится сложность XForms с XHTML.

Ну вот, приняли еще один хороший стандарт, и все усложнилось. Хоть кто-нибудь выиграл от этого?

О да, все это делается для удобства разработчиков, которые плюются при каждом упоминании об HTML. У них давно готовы web-сервисы, XML-схемы и коды J2EE, дело лишь за XForms – они сделают нашу Сеть много лучше.

Это все?

Нет, не все. Вместе с XForms, пожалуйста, встречайте XFrames, попытку замены устаревших фреймов, о необходимости которой говорили годами. Рабочая группа XFrames посвятила свое время решению проблемы людей, которых сбивает с толку поведение кнопки браузера «Назад». Широко известны также проблемы тех, кто не может поставить закладку на конкретную страницу, не может определить происхождение фреймовой группы и, что хуже всего, в этой фреймовой группе заблудился. Теперь им полегчает.

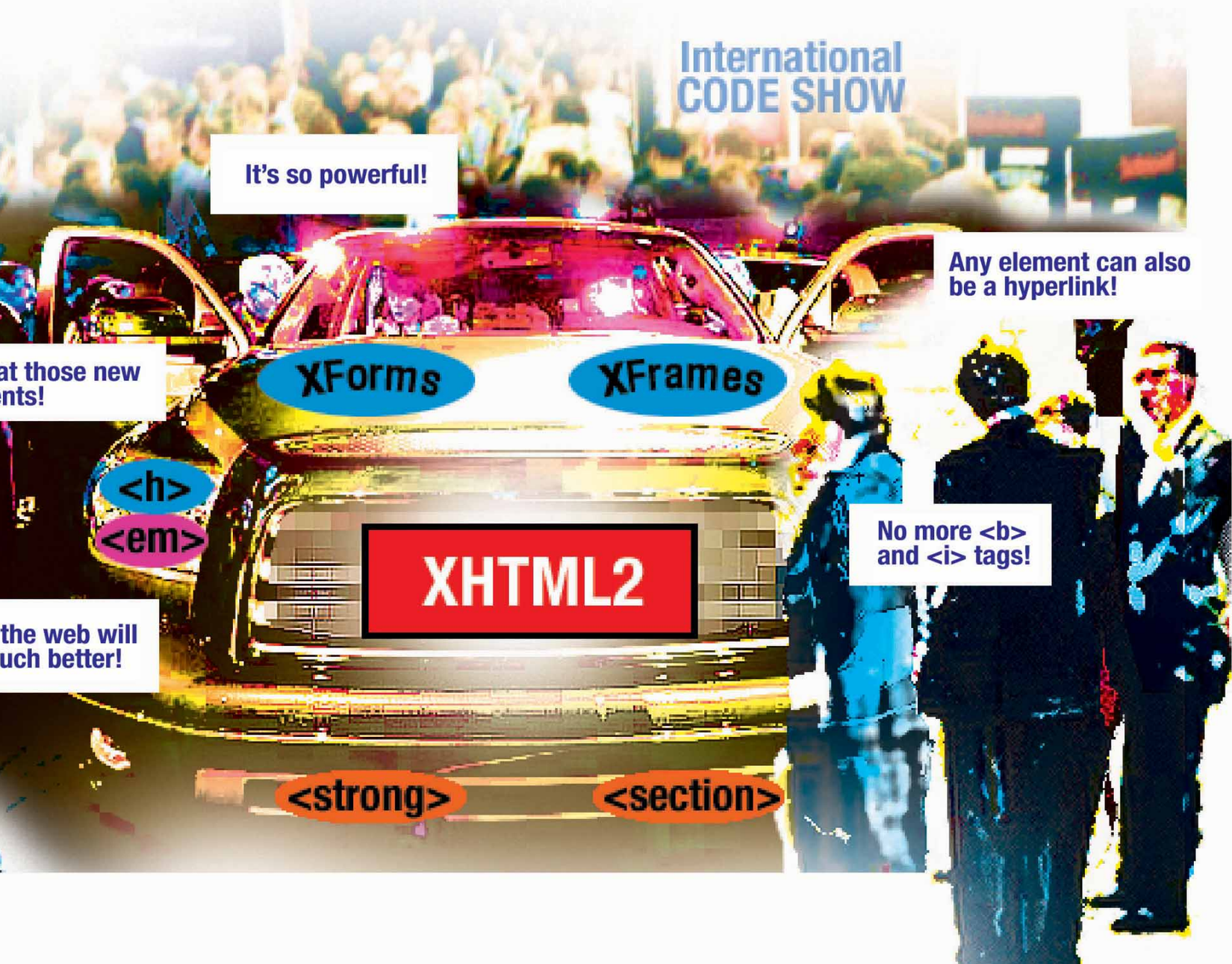
Спасибо; а где можно узнать подробности?

Эй, я ведь еще не закончил! Невозможно рассказать о XHTML 2 без упоминания о новых элементах, <h> и <section>. Они пришли на замену прежним <h1>, <h2> и им подобным, а уровень заголовка теперь определяется тем, в какой секции находится элемент <h>. Фактически, если говорить о URL, точно так же, как каждый элемент может содержать рисунок, каждый элемент может быть и гиперссылкой. Например:

```
<img src="tacheback.gif" href="http://tinyurl.com
yofm6c"><strong>Andrew Gregory</strong>
winking like a mad man</img>
```

Ну, теперь-то вы назовете адреса?

Информация о XHTML 2 очень разбросана, но последняя редакция этого стандарта доступна на www.w3.org/TR/xhtml2. Введение в этот документ даст общее представление тому, у кого есть лишь десять свободных минут: www.w3.org/TR/xhtml2/introduction.html. Если же нет и десяти минут, с основными положениями можно ознакомиться вот здесь: www.petefreitag.com/item/370.cfm. LXF



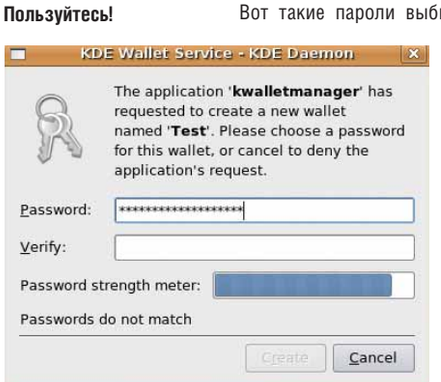
15 ценных советов по безопасности

Думаете, ваш компьютер с Linux защищен? Вы уверены? Даже самая безопасная ОС станет легкой добычей, если пренебрегать основными правилами. **Грэм Моррисон** даст 15 советов, чтобы предприимчивый взломщик плюнул на ваш ПК и убрался вон...

Усильте пароли пользователей

Знаем, это похоже на надоевшую проповедь. Кто хоть раз читал это, читал и тысячу раз: слабый пароль – самая частая «точка входа» потенциального взломщика. Но к проповеди все глухи. В октябре прошлого года 34 тысячи паролей пользователей MySpace [**myspace.com** – популярный сайт и сетевое сообщество – *прим. пер.*] были похищены в результате атаки фишеров, реализованной через поддельную страницу входа в систему. MySpace получил серьезную пробоину, а эксперт по безопасности Брюс Шнайер [Bruce Schneier] – прекрасную возможность заглянуть в коллективный разум 34-х тысяч типичных пользователей компьютера. Тремя самыми популярными паролями оказались password1, abc123, а самым креативным – myspace1. Шнайер ехидно отметил, что пользователи MySpace как минимум научились добавлять цифры к легко угадываемому слову, хотя все еще используют символы только в нижнем регистре.

Вот такие пароли выбираются типичными пользователями для защиты своих драгоценных данных. Если вы недостаточно напуганы, скачайте инструмент взлома паролей вроде *John the Ripper* (**www.openwall.com/john**) и натравите его на свой файл с зашифрованными паролями (см. LXF84). Шнайер рекомендует брать пароль, не включающий ни обычные слова, ни какие-либо производные сокращения, ни даты, которые легко узнать: все они подвержены атакам по словарю. Смешивайте регистры и добавляйте небуквенные символы.

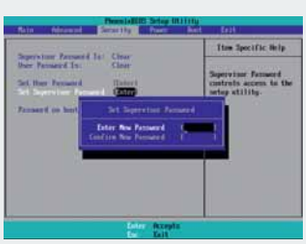


› Приложение *Wallet* из KDE включает измеритель силы паролей. Пользуйтесь!

Установите пароль BIOS

Не будем зацикливаться на паролях; это последнее упоминание о них. Но раз уж вы защитили свою учетную запись паролем, недоступным даже вашему двойнику из параллельной вселенной, почему бы не применить эту теорию к другим областям? Если вас беспокоит, что кто-то имеет физический доступ к вашей машине – например, она используется как сервер в небольшом офисе – установите стойкий пароль на BIOS вашей системы. Эта программа загружается первой, непосредственно отвечает за первичную настройку вашего оборудования и проверяет, что все работает должным образом.

Многие BIOS можно настроить на запрос пароля во время загрузки. Есть множество типов BIOS, и настройка будет зависеть от специфики вашего оборудования. Обычно, чтобы попасть в меню BIOS, следует удерживать [F2] во время загрузки машины, затем выбрать страницу Security, где вы сможете установить свой пароль. После этого единственным способом загрузить вашу систему, не зная пароля, будет очистка памяти BIOS – обычно тут не обойтись без вскрытия корпуса и извлечения батарейки материнской платы [не обольщайтесь – зачастую достаточно просто снять специальную перемычку; кроме того, некоторые версии BIOS принимают т.н. «инженерный пароль», – *прим. ред.*].



› Можете установить здесь пароль, чтобы никто не мог загрузить вашу машину.

Установите брандмауэр

Пару лет назад считалось чистым безумием подключаться к Интернету без брандмауэра. Правда заключается в том, что брандмауэры воспринимаются совершенно неправильно. Они лишь блокирует доступ к определенным сетевым портам на вашей машине. Если нет сервиса, прослушивающего этот порт, злонамеренный удаленный процесс не получит доступа к вашей системе независимо от того, запущен брандмауэр или нет. Именно поэтому Ubuntu не запускает брандмауэр по умолчанию: там просто не работают никакие сетевые сервисы, которыми мог бы воспользоваться злоумышленник. Брандмауэр может помочь только в том случае, если к вашей машине осуществляются как доверенные, так и не заслуживающие доверия сетевые подключения. В этом случае брандмауэр будет оберегать небезопасные сервисы от доступа из потенциально опасной сети. Уязвимыми сервисами могут быть внутренний web-сервер или общие ресурсы NFS.

› Графическая оболочка к брандмауэру, встроенная во все последние версии Linux, здорово экономит силы при настройке.

Функции брандмауэра в Linux фактически выполняются ядром, благодаря сложному инструменту *iptables*, используемому для фильтрации и перенаправления пакетов. Можно строить правила вручную, используя команду *iptables*, но большинство пользователей предпочитают графический интерфейс. Хороший выбор – *Firestarter*: он предлагает «мастера», помогающего выполнить стандартную настройку и даже разделить ваше интернет-соединение с другими компьютерами локальной сети. Вы всегда можете проверить эффективность защиты, используя для тестирования брандмауэра внешний сайт, например, *Shields Up* (www.grc.com/x/ne.dll?bh0bkyd2).

Разберитесь с правами

Многие из нас в курсе, что Unix-подобные ОС вроде Linux используют права для предоставления доступа к файлам и приложениям, но как убедиться, что они настроены должным образом? Взгляните на содержимое своего каталога **/home**, набрав в командной строке **ls -l /home**. Вывод *ls*, вероятно, будет подобен этому:

› Если вам сложно изменить права доступа из командной строки, попробуйте файловый менеджер Gnome или KDE.

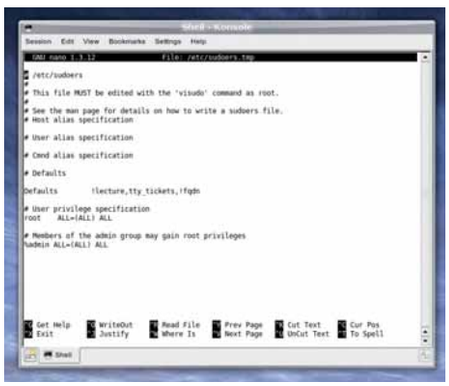
```
drwxr-xr-x 66 graham graham 2904 2007-01-23 10:09 graham
```

В выводе будет запись для каждого пользователя вашей системы, и проблема заключается в строке прав доступа – drwxr-xr-x. Если коротко, то d – это признак каталога, rwx – права доступа владельца файла (чтение, запись и исполнение), первые символы r-x – права группы, а вторые r-x – права всех остальных. Следующая часть, graham graham, обозначает пользователя и группу, владеющих этим каталогом. Многие дистрибутивы глобально открывают домашние каталоги пользователей на чтение, а значит, каждый, имеющий доступ к вашей машине, сможет прочитать содержимое вашего домашнего каталога. Вам следует изменить права с помощью chmod. Например, **chmod go-rx /home/graham** отнимет доступ на чтение и исполнение у всех пользователей в этой же группе, а также у всех прочих пользователей.

› На стр. 56 представлено руководство по правам доступа для начинающих.

Следите за sudo

Для быстрого запуска приложений с привилегиями администратора системы в Ubuntu применяется утилита *sudo*. Именно для этого *sudo* и разработана. Но предоставление любому пользователю доступа через *sudo* может быть рискованно с точки зрения безопасности, поскольку вы не контролируете, кто и что запускает. Решение – разрешить использование *sudo* только конкретным пользователям и точно настроить, какой доступ и к каким средствам им необходим.



› Мощная команда sudo легко настраивается с помощью встроенного редактора *visudo*.

Для редактирования привилегий *sudo* существует файл конфигурации, но изменять его нужно собственным *vi*-подобным редактором под названием *visudo*. Первое, что нужно сделать – это отредактировать строку root ALL=(ALL) ALL, заменив root на список пользователей (через запятую), которых вы хотите допустить к командам, требующим повышенных привилегий. Другой частый пример – заменить секцию после 'ALL=' на список каталогов (тоже через запятую), к которым вы хотите предоставить доступ перечисленным пользователям: два наиболее типичных случая – **/sbin** и **/usr/sbin**. Чтобы подпускать к учетной записи root только пользователей – членов группы admin, введите следующее:

```
%admin ALL=(ALL) ALL
```

Не используйте setuid – правда!

Setuid – прекрасная практическая идея: это просто еще один бит прав доступа, наряду с чтением, записью и исполнением, позволяющий запускать команду с привилегиями владельца файла. То есть обычный пользователь может выполнять команды администрирования, если привилегии, требуемые для их выполнения, определены правом setuid. Setuid также полезен для приложений, работающих с оборудованием, доступным только администратору: например, приводов CD-RW. Но применение setuid создает значительный риск. Если использующая его команда по той или иной причине «рухнет», запустившему ее пользователю могут достаться привилегии «погибшей» программы. Плохие ребята, взламывающие Linux, точно знают, какие setuid-инструменты можно уговорить вести себя неправильно.

Хороший пример использования setuid – команда *ping*, требующая административных привилегий для получения доступа к сети. Команда **ls -l /bin/ping** выдаст следующее:

```
-rwsr-xr-x 1 root root 30804 2006-10-16 18:32 /bin/ping
```

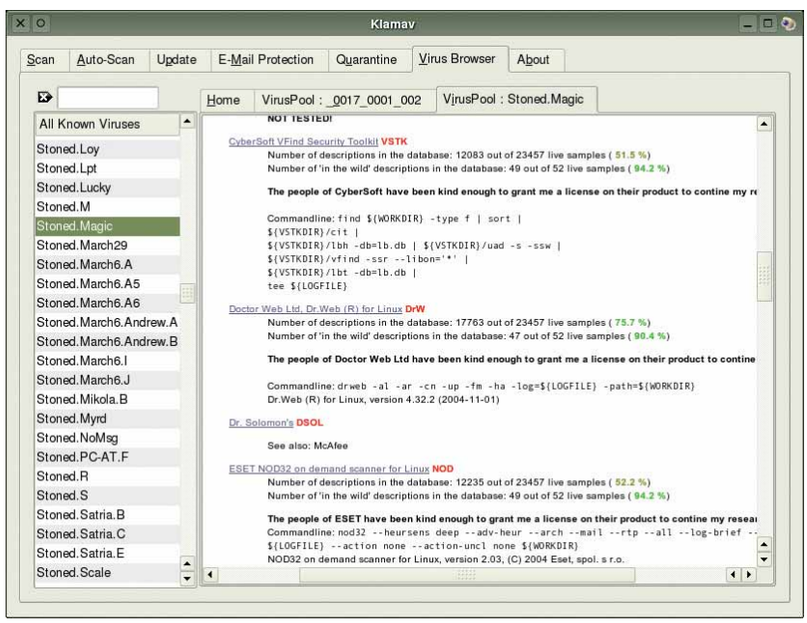
Дополнительный символ s в секции прав показывает, что бит setuid установлен. Во время выполнения *ping* получает права пользователя-владельца файла, в данном случае это root. Оставить инструменты с установленным setuid в вашем дистрибутиве – приемлемый риск, но вы должны дважды подумать, прежде чем использовать его для упрощения жизни при написании своих собственных скриптов.

Запускайте проверку на вирусы

Хотя Linux не слишком обременен разрушительными вирусами, все же сравнительно легко непреднамеренно пропустить вирус к тем, с кем вы взаимодействуете, потому что на вашем рабочем столе могут потихоньку дремать зараженные файлы – плод злонамеренного почтового сообщения или скачивания. Даже если они не в силах сделать ничего дурного на вашей Linux-системе, они по-прежнему способны »

» причинить вред какой-нибудь другой ОС. Запуск антивирусного ПО на вашем Linux-компьютере даст вам душевное спокойствие при отправке файлов вашим знакомым, а также отловит подозрительные двоичные файлы, не позволив им покинуть лабораторию.

Наш любимый антивирусный пакет – *ClamAV*, имеющий версии для Mac и Windows, если вы хотите соблазнить друзей открытым ПО. Он полностью бесплатен, и база данных известных вирусов обычно обновляется в течение 24 часов с момента обнаружения нового вируса. *ClamAV* выиграл наше сравнение антивирусов в LXF74/75, поскольку он достаточно гибок для использования как на почтовом сервере, так и в настольной системе. Если он установлен, вы можете проанализировать любые подозрительные файлы, просто щелкнув на файле правой кнопкой мыши и выбрав пункт «сканирование на вирусы».



› ***ClamAV*** **можно запустить из командной строки, или, если хотите, через графический интерфейс для KDE или Gnome.**

Закалите SSH

Теперь, когда вы сделали пароли более стойкими, захватить машину удаленно – через SSH – будет куда как труднее. Но вы можете сделать еще больше, чтобы быть уверенным, что какой-нибудь пацан с другого континента не покопается в ваших личных файлах. Первое – это запретить удаленный доступ пользователю root в конфигурационном файле SSH. Откройте **/etc/sshd.conf** или **/etc/ssh/sshd_config** в своем любимом текстовом редакторе и убедитесь, что опция PermitRootLogin установлена в **no**. Теперь вам, а заодно и пацану с другого континента, нужно будет получать доступ с правами root через учетную запись обычного пользователя.

Еще одна опция, которую стоит поискать в файле конфигурации – Protocol 2. Любые другие строки с 'Protocol' нужно закомментировать, поскольку они разрешают соединения, использующие менее безопасные ранние версии протокола, лишая вас при этом множества преимуществ.

Заприте SSH на ключ

Самый безопасный способ работать с SSH – это использовать для аутентификации соединения пару из открытого и закрытого ключей и запретить доступ к SSH-серверу с обычным паролем. Вы можете сгенерировать ключи на клиенте с помощью **ssh-keygen -t dsa**. У вас будет запрошено месторасположение ключа (его можно оставить по умолчанию), а затем парольная фраза. Если вы оставите парольную фразу пустой, при создании нового SSH-соединения никакие пароли запрашиваться не будут, и соединение будет установлено прозрачно. Это полезно для автоматизации таких задач, как удаленное резервное копирование, но каждый, имеющий копию этого ключа, сможет подключиться к вашему серверу.

Следующий шаг – скопировать открытый ключ на сервер. Это выполняется командой

```
ssh-copy-id -i ~/.ssh/id_dsa.pub server_address
```

Теперь вы должны получить возможность подключаться к своему SSH-серверу, используя аутентификацию по ключу.

Если вы столкнетесь с проблемами, первым делом убедитесь, что аутентификация по открытому ключу разрешена в файле конфигурации SSH (это выглядит как PubkeyAuthentication yes), а если это не поможет, убедитесь, что и ваш домашний каталог, и каталог **.ssh** на удаленной машине имеют права доступа 'drwx------'. Наконец, когда все заработает, запретите в конфигурационном файле SSH аутентификацию по паролю.

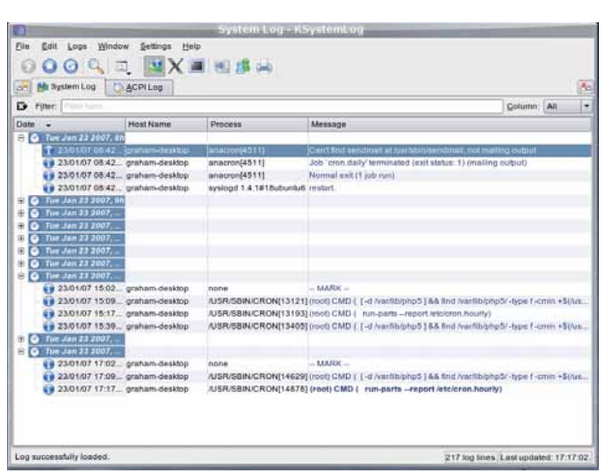
Раздавайте свой открытый ключ

Если вы приедете на одну из крупных конференций разработчиков открытого ПО, например, OSCon, Akademy или Guadec, вы, вероятно, увидите странный пункт в программе мероприятия, называющийся 'key signing party' («вечеринка подписания ключей», не путайте с 'key party' [key party – вечеринка, в начале которой мужчины складывают ключи от машин в кучу, а в конце женщины выбирают случайный ключ, определяя тем самым своего сексуального партнера, – *прим. пер.*]). Это возможность для разработчиков поделиться открытыми PGP-ключами, а также единственный способ получить на 100% достоверный ключ, не перехваченный при пересылке (что известно как «атака посредника»). Пусть вы не разработчик, но распространение своего открытого ключа по доверенному каналу – прекрасный способ подтвердить свою личность, во многом подобный подписи в реальной жизни.

Сперва вам нужно создать ключ, набрав **gpg --gen-key**. Выберите опции по умолчанию с размером ключа '4096' для параноиков и введите свой адрес электронной почты. Важнейшая часть – ключевая фраза, она используется для защиты вашего ключа таким же способом, как пароль защищает учетную запись. Чем длиннее и случайнее по своей природе эта фраза, тем сложнее ее взломать. Вы должны будете отправить идентификатор своего ключа на электронный адрес, прежде чем получить список всех ключей, которые собираетесь подписывать.

На самой вечеринке вы можете удостоверить личности людей в списке и подлинность их ключей. Затем вы отправляете их ключи к своему и отсылаете копию партнеру, который поступает так же – тем самым будет построена так называемая «сеть доверия» между участниками. PGP-ключи также можно использовать для шифрования файлов и электронных сообщений, предназначенных к отправке через небезопасную сеть. Ими гораздо проще управлять, когда используются графические интерфейсы типа *Kpgp*, или когда они интегрированы в почтовый клиент типа *Evolution* или *KMail*.

Проверяйте системные журналы



› ***KSysLog*** **рассортирует ваши системные журналы по вкладкам и категориям и подсветит записи, требующие внимания.**

Linux – одна из самых разговорчивых ОС. В журналах можно найти запись почти для каждой возникающей проблемы. Возможно, именно по этой причине очень немногие из нас не ленятся их просматривать. Что довольно глупо, поскольку файлы журналов могут показать, как атаковалась ваша система и какие меры следует предпринять, чтобы атаки терпели поражение.

Все системные файлы журналов размещаются в каталоге **/var/log**, и можно многое узнать, просто просматривая их время от времени. Файл с именем **messages** – хорошее место для старта. Он содержит общие сообщения системы, а также протокол загрузки. Протокол последней загрузки также можно отобразить командой *dmesg*. Если вы хотите найти что-нибудь определенное, вам поможет *grep*. Используйте команду *tail -f*, чтобы просматривать на экране последние записи в журнале и команду *logger* для добавления собственных сообщений.

› Читайте о ведении журналов на стр. 38.

Будьте на шаг впереди

Когда вы обнаружите проблему, не тяните с решением. В случае спама для поимки виновного мало что можно сделать. Но в Интернете можно найти обратный адрес отправителя, а также машину, использованную для отсылки сообщения, и добавить ее адрес в базу данных своего спам-фильтра (вы же используете спам-фильтр, не так ли?). А вот об атаках фишеров следует сообщать. Во-первых, организации, которая оказалась под атакой, а во-вторых, на сайт, отслеживающий фишеров, например, www.antiphishing.org.

Другой способ активной защиты: если у вас работает сервер SSH, приглядывайте за файлом **/var/log/auth.log**, и рано или поздно увидите, что кто-то пытался угадать ваш пароль, чтобы дорваться до вашей машины. Строка в журнале может выглядеть, например, так:

```
Failed password for invalid user test from ip_address port 55218 ssh2
```

Здесь кто-то пытался соединиться с вашей машиной как пользователь test, но потерпел неудачу. Вы узнаете больше о компьютере, который покушался на вашу машину, набрав **whois ip_address**; вывод зачастую включает номер контактного телефона. В большинстве случаев это окажется обычный провайдер, который более чем заинтересован в сведениях, что один из его клиентов пытался получить неавторизованный доступ к чужой системе.

Защищайте беспроводную сеть

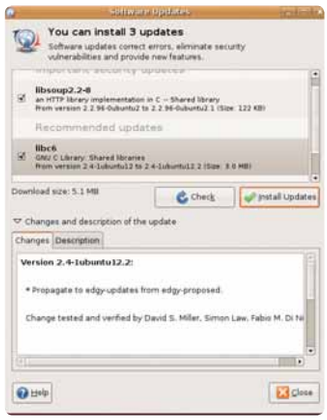
Достаточно открыть ноутбук и просканировать точки доступа в густонаселенном районе, чтобы увидеть: существует масса людей, использующих незащищенные беспроводные сети. Помимо проблемы расходования дорогого канала халявщиками, незащищенная сеть неприлично проста для слежки. Если не применяется шифрование, ничто не помешает любопытствующему читать вашу почту и мгновенные сообщения как бесплатную газету.

Шифрование поможет в этой беде. Есть две опции, поддерживаемых среднестатистической точкой доступа: WEP и WPA. WEP старше, и имеет известные уязвимости. Если кто-то стянет достаточно данных, зашифрованных с помощью WEP, из этих данных вполне осуществимо извлечь ключ, и ваше соединение будет взломано. WPA, напротив, сейчас считается безопасным. Единственная проблема WPA заключается в том, что в Linux беспроводные драйверы только начали поддерживать его.

Оба метода работают, в общем, одинаково. Вы задаете пароль для точки доступа и используете его на вашем ноутбуке для подключения к сети. Благодаря современным инструментам Linux, таким, как *Network Manager* в Gnome и его KDE-эквивалент, управление паролями выполняется весьма просто. Как только пароль будет принят, вы сможете свободно использовать свое соединение.

Поддерживайте актуальность

На данный момент у вас могла возникнуть мысль, что определенные экземпляры ПО в Linux уязвимее других. Чем больше программ у пользователя, тем вероятнее, что будет обнаружена уязвимость или ошибка. Все, что имеет выход в Интернет, всегда подпадает под больший риск, чем небольшая программа с малым количеством доверенных пользователей, и поэтому проблемы с пакетными версиями *Apache* и PHP являются вполне обычными. Чтобы контролировать эти проблемы, следует регулярно обновлять свою систему. Многие дистрибутивы Linux сейчас включают менеджер обновлений, который доставляет их на вашу систему по мере обнаружения уязвимостей. Вы должны убедиться, что это действительно происходит. В частности, в Ubuntu слишком легко отключить *Update Manager* и забыть запустить его вновь. Это очень печально, поскольку для 6.06 LTS обещаны обновления безопасности в течение пяти лет.



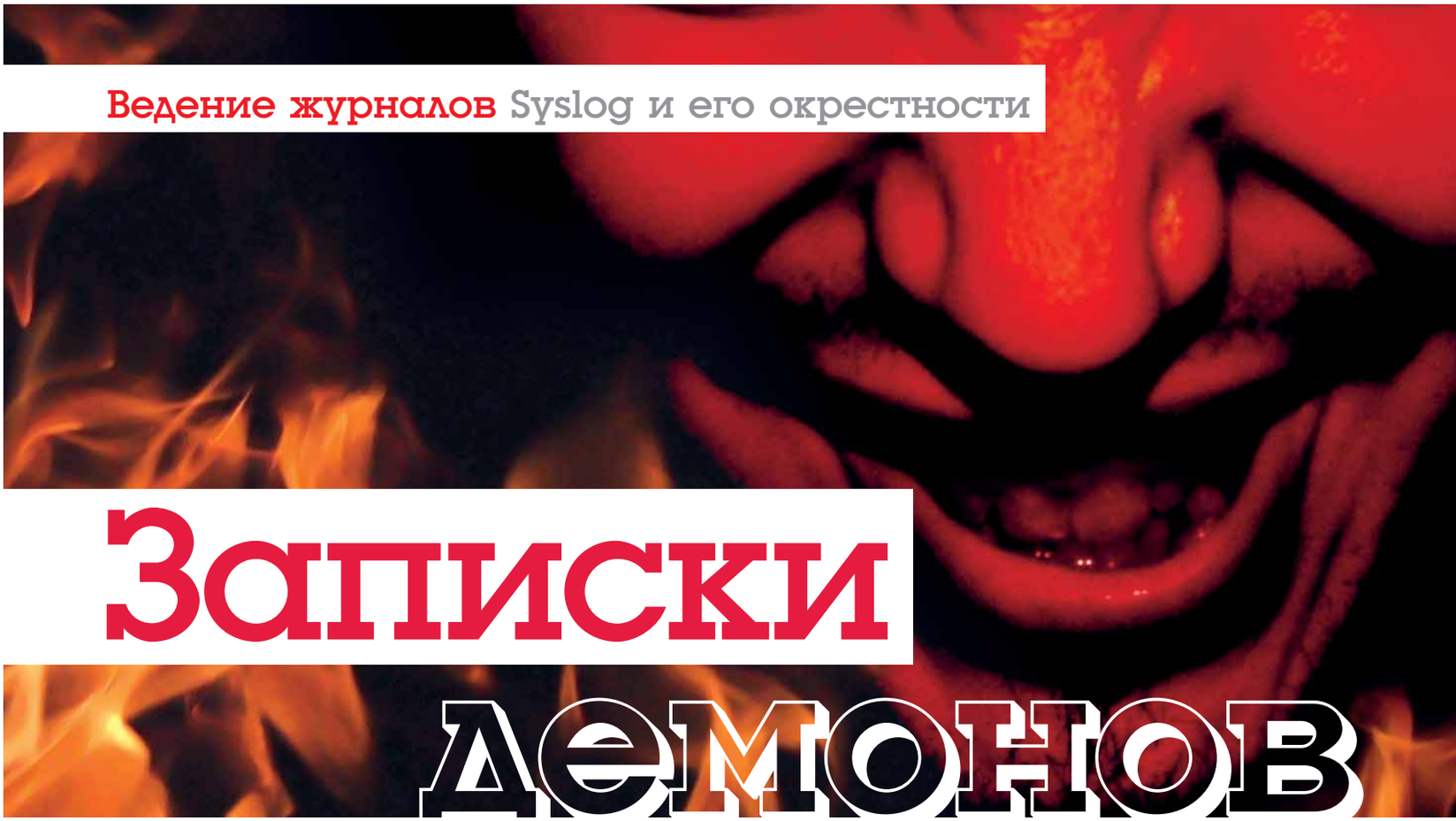
› Будьте бдительны: отключив инструмент автоматического обновления своего дистрибутива, вы пропустите важные обновления безопасности.

Учтите, безопасность не приходит сама

Это самый важный совет. В Linux нет ни ежедневных напоминаний, от которых страдают пользователи других операционных систем, ни финансового бремени от запуска платных антивирусов, брандмауэра и блокировщика всплывающих окон. Но это не значит, что нет угрозы. Если вы не поленитесь заглянуть в свои файлы журналов, то увидите, что она определенно есть. И когда система, которую вы считали безопасной, взламывается, последствия могут быть вдесятеро ужаснее, чем просто заражение вирусом. Ваша машина может использоваться для размещения нелегального ПО, организации атак типа «отказ в обслуживании», или просто для рассылки тысячи спам-сообщений в час.

Простое следование некоторым из наших советов определенно поможет, но то же самое достигается обычной осторожностью. Например, подумайте дважды, прежде чем устанавливать неподписанные программы из сторонних репозиториев. Даже подписанные файлы не обязательно легитимны. Прикиньте, как легко встроить маленькую программку в исходный код одного из тех пакетов, которые вы скачиваете, и сколько разрушений она может понаделать, отсылая историю ваших действий на какой-нибудь анонимный сервер в Интернете. Шансы невелики, но лучше быть готовым к худшему. При наличии выбора, взломщик перейдет к более легкой цели. LXF

› Программа управления пакетами должна предупреждать вас о неподписанных и сомнительных пакетах.



Записки демонов

Учитесь правильно читать файлы журналов: они дают ценные данные о вашей системе. **Д-р Крис Браун** начинает серию руководств из двух частей с рассказа о *syslogd*.

Не будем врать: файлы журналов, вероятно, скучнейшие из всех в Linux-системе. Они однообразны и нудны, и просматривать их – все равно что перематывать бесконечные видеозаписи службы безопасности в поисках чего-нибудь интересного. Вдобавок сообщения в журнале часто пишутся в расчете на разработчика, а не конечного пользователя, и понять их трудно.

Логично спросить: зачем вообще смотреть файлы журналов? Никуда не денешься: файлы журналов рассказывают о сервисах системы (демонах) и других программах, которые не имеют видимого пользователю интерфейса и не могут сами рассказать о своих действиях и ошибках. Файлы журналов – это записки демонов.

Например:

1 Файлы журналов сообщат, насколько загружен сервер. Допустим, вам нужно выставить счет за его использование или понять, хорошо ли сервер работает как средство маркетинга или доставки данных. Журналы web-сервера особенно важны, и существует довольно много инструментов, позволяющих выдавать статистику на основе файлов журналов сервера *Apache*.

2 Файлы журналов помогут выявить ошибки в настройках (например, неправильные настройки авторизации) или отсутствие файлов (например, ошибки типа '404 файл не найден').

3 Файлы журналов прояснят, почему сервис не желает правильно запускаться. Это особенно ценно при первом запуске приложения после изменений в настройке. Мудрые администраторы запускают **tail -f** на файле журнала (тогда можно просматривать файл по мере его роста) в одном окне терминала, а в другом запускают сервер.

4 Файлы журналов расскажут, что кто-то норовит вломиться в вашу систему. Фактически, о любой машине, имеющей внешний видимый IP-адрес, можно утверждать, что кто-то пытается в нее проникнуть. Вопрос, преуспел ли этот кто-то? К примеру, журнал сервера под управлением автора содержит свыше 50000 строк, относящихся к попыткам проникновения – и это только за одну неделю!

В данной серии из двух уроков мы хотели бы помочь вам понять и настроить процесс регистрации событий. Начнем с создания файлов журналов, затем разберемся, как и где настроить журналируемые события. А через месяц рассмотрим некоторые инструменты для управления, анализа и обобщения этих файлов.

Тонкости Syslogd

Нет единого жесткого правила, определяющего, что нужно записывать. По сути, сервис записывает сообщение при совершении действия, которое создатель программы посчитал достойным упоминания. FTP-сервер может создавать запись каждый раз при запросе файла; ядро – находя новое устройство; и т.д., причем обычно стараются записывать события, выходящие за рамки обычных.

Некоторые сервисы, например, *Apache*, ведут свои собственные журналы. Другие – включая почту, печать, подсистему безопасности, *Cron* и ядро – делают записи с помощью отдельного демона, *syslogd*, обрабатывающего сообщения от их имени. Рассмотрим сначала метод syslog.

Отправляя записи через *syslogd*, сервисы не только передают тексты сообщений, но и указывают 'источник' (facility) и 'уровень' (level). Источник идентифицирует подсистему, от которой пришло сообщение, а уровень означает его важность. Syslog имеет файл настроек, определяющий, куда посылать то или иное сообщение, на основе его источника и уровня (как мы увидим, записи не обязательно направляются в журнал, хотя это их обычный путь).

Список источников включает auth, authpriv, cron, daemon, fpm kern, lpr, mail, news, syslog, user, uucp и от local0 до local7. Восемь источников local *syslogd* предоставляет для пользовательских нужд. Кому интересно, uucp означает 'Unix to Unix copy', это древний набор программ для удаленной передачи файлов и выполнения программ. Название также выдает возраст syslog: он начал использоваться с 1980-х.

Существует восемь возможных уровней, начиная от щадящего до катастрофического, как показано в таблице *«Уровни Syslog»* справа вверху. (Описание каждого уровня является нашей интерпретацией). Между прочим, некоторые авторы используют термин 'приоритет', а не 'уровень', но большая часть документации по syslog использует термин 'приоритет' для обозначения комбинации источника и уровня. Будьте осторожны – возможны недоразумения.

Что происходит, когда сообщение доходит до *syslogd*? Это зависит от файла настройки, но возможны пять вариантов:

- Оно может быть добавлено в файл. Это наиболее распространенный выбор.
- Оно может выдано на терминал любого указанного пользователя.
- Оно может быть записано в FIFO (именованный канал). Это бывает полезно при отладке; или можно запустить grep и вытаскивать интересные сообщения из FIFO, пользуясь шаблоном регулярного выражения.
- Оно может быть перенаправлено *syslogd*, находящемуся на удаленном узле.
- Наконец, если для сообщения не определено, что с ним делать, оно просто игнорируется.

Мы скоро рассмотрим каждое из этих действий подробно. А сейчас займемся самым важным файлом настройки, **/etc/syslog.conf**, который связывает все вместе. Вот возможные варианты строк этого файла. Это не настоящие настройки, просто набор примеров для пояснения синтаксиса. Номера строк даны для удобства ссылки – в сам файл они не входят.

```
1   mail.err                    /var/log/mail
2   mail.*                      /var/log/mail
3   mail.debug                  /var/log/mail
4   *.crit                      /var/log/critical
5   *.*                         @loghost
6   mail.=debug                 /var/log/maildebug
7   mail.warn;cron.notice       var/log/messages
8   *.*;auth.none               /var/log/messages
9   auth,kern.crit              /var/log/critical
10  *.*;auth,kern.none          /var/log/messages
11  *.=debug;*.=info            -/var/log/messages
12  *.crit                      root
13  *.crit                      *
14  *.=notice;*.=warn           l/dev/xconsole
```

Каждое правило содержит селектор и действие. Так, в строке 1 селектором является mail.err. Это значит, что правило применяется к сообщениям от источника mail уровня err или выше; то есть уровни err, crit, alert или emerg. Затем идет действие – добавить сообщение в файл **/var/log/mail**. Легко, правда?

Правила бывают и посложнее. Селекторы допускают символы подстановки (*) как для источника, так и для уровня. Так, селектор в строке 2 означает 'все сообщения от источника mail' согласно принципу 'от этого уровня и выше', строка 3 делает то же самое. Селектор в строке 4 означает 'сообщения уровня crit (или выше) от любого источника', а строка 5, понятное дело, применяется ко всем сообщениям. Знак равенства (=) перед уровнем означает, что правило применимо только к этому уровню, поэтому правило в строке 6 применимо сообщениям от источника mail только уровня debug. Можно указать несколько селекторов, разделив их точкой с запятой(;) как показано в строке 7 (такой же эффект достигается написанием двух отдельных правил). Пустой уровень none используется для исключения всех сообщений от данного источника и обычно используется вместе с ;, как показано в строке 8, соответствующей всем сообщениям, кроме идущих от источника auth.

Новое поколение

Некоторые из последних дистрибутивов – отметим SUSE – заменили *syslogd* на *syslog-ng*. Этот демон обратно совместим с *syslogd* (он все еще использует источники и уровни), но дает системному администратору больший контроль над тем, откуда приходит сообщение и куда оно пересылается (ценой усложнения файла настройки). Сообщения могут выбираться на основе регулярных выражений, и для удаленного журналирования используется TCP, а не UDP. Если вы хотите, чтобы мы рассмотрели *syslog-ng* подробно, пишите на letters@linuxformat.ru.

Уровни Syslog

Сообщениям журнала присваивается «уровень», отражающий степень их важности:

Уровень	Описание
debug	Отладочные сообщения. Рассказывают о внутренней деятельности сервиса, обычно велики по объему и часто нацелены на разработчиков, а не пользователей. Чтобы увидеть сообщения, связанные с отладкой, многие демоны необходимо запускать с определенными флагами и/или собирать с определенными ключами.
info, notice	Эти сообщения регистрируют обычную (но важную) деятельность сервиса.
warning	Необычные или ненормальные события, которые, однако, не вызывают сбоя сервиса.
err	Ошибки, способные вызвать отказ сервиса при ответе на запрос.
crit, alert, emerg	Эти три уровня представляют ошибки (в порядке возрастания серьезности), способные привести сервис (или всю систему) к полному отказу, и требуют немедленной реакции.

Наконец, если надо, чтобы селектор включал несколько источников одного уровня, отделите имена источников запятой (,) как показано в строке 9. Между прочим, для сообщения вполне нормально соответствовать более чем одному селектору – *syslogd* просто выполнит все предписанные действия, по очереди.

Предпринимаем действие

Как мы уже упомянули, чаще всего сообщения добавляются к файлу; вы просто определяете в качестве действия (абсолютный) путь к нему, как мы делали в наших примерах. Обычно *syslogd* сбрасывает свои буферы на диск после каждой записи. Это увеличивает шансы сообщения попасть в файл до того, как система рухнет, но это также значит, что менее критичные (и более объемные) сообщения уровней debug, info и notice вызывают излишнюю дисковую активность. Поставив

> «Ведение журналов в сети легко централизовать, создав для них отдельный хост.»

дефис (-) перед именем файла, вы разрешите *syslogd* не сбрасывать буферы на диск каждый раз (см. строку 11). Можно попросить сообщение отобразиться на консоль любого подключенного пользователя (root является фаворитом), определив в качестве действия имя учетной записи, как в строке 12.

Здесь также применяется символ подстановки (*); действие в строке 13 означает 'написать всем подключенным пользователям'. Во времена, когда системный администратор постоянно сидел в текстовой консоли (если такое вообще было), это имело значение, но настольные компьютеры работают в графическом режиме, а за серверами особо не присматривают. Вы можете заставить *syslogd* перенаправлять сообщения на удаленную машину, добавив знак @ перед именем машины, указанным в качестве действия; пример приведен в строке 5, но мы подробно рассмотрим его попозже.

Наконец, можно велеть *syslogd* записывать сообщения в именованный канал, поставив символ канала (|) перед его именем; пример – строка 14 (взятая из стандартного **syslog.conf** в Ubuntu).

Поэкспериментируем

В порядке иллюстрации, настроим *syslogd* так, чтобы он посылал все сообщения от источника local6 в файл **/var/log/daemon**. Для внесения изменений необходимо быть суперпользователем. Добавьте в файл **syslog.conf** строчку: »

```
local6.notice           /var/log/demolog
```

Далее, из командной строки, пошлите *syslogd* сигнал SIGHUP, чтобы он перечитал файл.

```
# pkill -HUP syslogd
```

Для отправки сообщения в *syslogd* из командной строки служит команда logger. Вот типичный пример ее использования (опция -p указывает на источник и уровень сообщения):

```
# logger -p mail.info “Тестовое сообщение от источника mail”
```

Чтобы послать сообщение с созданным нами приоритетом local6.notice, выполните

```
# logger -p local6.notice “Это тест”
```

Теперь просмотрите файл **/var/log/demolog**. Там должна быть примерно такая строка:

```
Dec 27 10:38:38 frodo chris: Это тест
```

Вы увидите, что *syslogd* предварил сообщение некоторой информацией: в данном случае это отметка времени, имя машины и UID процесса, пославшего сообщение. Попробуйте записать сообщения от источника local6 с различными уровнями и проверить, какие уровни записываются.

Если посылать одно и тоже сообщение *syslogd* много раз подряд, то *syslogd* будет сохранять их раз в минуту и добавлять отметку вроде ‘last message repeated 22 times’ [«последнее сообщение повторялось 22 раза»] в конце каждого интервала времени. Это не дает демонам распоясаться и затопить файлы журналов потоком однотипных сообщений.

Централизация журналов

По умолчанию, *syslogd* прослушивает Unix-сокет **/dev/log**, то есть доступен только процессам на локальной машине. Однако он также может прослушивать и UDP-сокет, и, как мы уже видели, одним из действий syslog является перенаправление сообщения в *syslogd* на удаленной машине. Это позволяет централизовать подсистему журналирования в вашей сети, выделив одну машину полностью под работу с журналами, чтобы другие просто пересылали ей свои сообщения. У такого подхода есть несколько преимуществ. Первое, сбор журналов на одной машине упрощает их анализ. Второе, это более безопасно. Если журналы хранятся на машине локально, коварный нарушитель может отредактировать их и замести свои следы; а если журналы хранятся на другой машине, то до нее надо еще добраться.

Однако с точки зрения безопасности удаленное журналирование имеет и недостаток: уязвимость к атакам на отказ в обслуживании от тех, кто просто посылает лавину сообщений в *syslogd*, пока не забьет весь диск. Поэтому имеет смысл установить на узле, ведущем журнал, брандмауэр и принимать пакеты на порт syslog (UDP-порт 514) только из своей локальной сети.

Вы заметите, что *syslogd* очень гибок в настройке обработки сообщений. Сложите все сообщения в один файл, разбейте их по нескольким файлам или просто передайте их удаленной машине – выбор за вами. На своих системах вы найдете различные вариации файла **syslog.conf**. Одно из преимуществ использования *syslogd* – для изменения стратегии ведения журнала достаточно отредактировать один файл.

Взгляд разработчика

Прежде чем покинем syslog, взглянем на журналирование с точки зрения разработчика демона (не путать с заклинателями демонов). Сообщения в *syslogd* легко посылать из программы на C.

```
#include <syslog.h>
#include <fcntl.h>
int main()
{
    openlog(“mydaemon”, LOG_PID, LOG_LOCAL6);
    if (open(“/etc/xyzzy”, O_RDONLY) < 0) {
        syslog(LOG_NOTICE, “xyzzy: %m”);
    }
    return 0;
}
```

Первый аргумент openlog – это идентификатор, который будет появляться во всех сообщениях (обычно имя демона); аргумент LOG_PID велит включать в сообщение идентификатор процесса демона;

Шаг за шагом: Удаленное журналирование

1 Обновим файл настройки

Давайте сделаем так, чтобы сообщения приоритета local6.notice журналировались удаленно. Вам потребуется две Linux-системы (назовем их «машина A» и «машина B») и плитка шоколада. На машине A добавим запись в **/etc/syslog.conf**:

```
local6.notice           @loghost
```

2 Добавим удаленный IP адрес

Все еще на машине A, добавим в **/etc/hosts** что-то наподобие

```
192.168.0.14            loghost
```

Подставьте сюда IP-адрес машины B.

3 Запустим pkill

Теперь прикажите *syslogd* перечитать файл настройки с помощью команды

```
# pkill -HUP syslogd
```

4 Настроим syslogd

На машине B надо удостовериться, что *syslogd* запущен с опцией -r так, чтобы он прослушивал UDP-порт. На системе с Fedora, например, потребуется отредактировать **/etc/sysconfig/syslog**, включив строку:

```
SYSLOGD_OPTIONS=”-m 0 -r”
```

5 Перезапустим демон

На машине B перезапустите *syslogd*. В Fedora или Red Hat это можно сделать командой

```
# service syslog restart
```

В Ubuntu, где нет *sysconfig*, понадобится отредактировать скрипт загрузки **/etc/init.d/syslogd** и добавить флаг -r в определение переменной *Syslogd*. Затем нужно ввести команду

```
# /etc/init.d/sysklogd restart
```

для перезапуска демона.

6 Проверим сокет

На машине B запустите команду netstat -au и проверьте, что на порте syslog открыт активный UDP-сокет.

7 Зададим местоположение

Далее отредактируйте **/etc/syslog.conf**, определив, куда должны идти сообщения от local6.notice. Например, добавьте строку

```
local6.notice /var/log/demolog
```

Все еще на машине B, прикажите *syslogd* перечитать файл, как вы сделали это на машине A в шаге 3.

8 Проверим соединение

Теперь все установлено. Проверьте настройку, запустив на машине A:

```
# logger -p local6.notice “Тестирование удаленного журналирования”
```

9 Убедимся, что оно работает

На машине B проверьте файл **/var/log/demolog** и убедитесь, что сообщение прибыло. Если да, поздравляем: удаленное журналирование готово! Если вы еще потихоньку не съели свой шоколад, можете сделать это сейчас. Нам он больше не понадобится. Заметим: если удаленное журналирование отказывается работать, проверьте, пропускает ли брандмауэр трафик *syslogd* на каждой из машин. В установках Linux по умолчанию он, скорее всего, заблокирован.

а последний аргумент указывает, что сообщения будут поступать от источника local6. (Эти символические константы определены в файле **syslog.h**.) Вызов syslog() посылает сообщение: первый аргумент – это уровень, а второй аргумент – строка в формате printf, определяющая оставшийся текст сообщения. В примере показан особый код формата – %m, он генерирует текст, описывающий последнюю ошибку – в данном случае, отказ вызова open() строкой выше. По выполнении программы, в нашем демо-журнале появится результирующая строка:

```
Dec 27 19:40:55 frodo mydaemon[22572]: xyzzy: No such file or directory
```

Чтобы регистрировать сообщения из сценария на языке оболочки, конечно, используется команда logger, изученная нами ранее.

Путь Apache

Syslog – не единственный способ ведения журналов. Некоторые сервисы – в частности, *Samba* и *Apache* – делают все сами. Где находятся эти журналы и что именно в них пишется, определяется в собственном файле настройки сервиса. «Место жительства» файла настройки *Apache* варьируется от дистрибутива к дистрибутиву; в Fedora, например, это **/etc/httpd/conf/htppd.conf**.

Apache обычно ведет два журнала: журнал передач и журнал ошибок. Вообще-то журнал ошибок создается независимо от того, просите вы об этом или нет, но вы можете явно определить его расположение директивой ErrorLog; например:

```
ErrorLog /var/log/Apache/errorlog
```

Вы можете также попросить *Apache* вести журнал через *syslogd*, например:

```
ErrorLog syslog:local2
```

где local2 – источник, от имени которого будет вестись журнал syslog.

Директива LogLevel поможет вам управлять уровнем подробностей в журнале ошибок. Например,

```
LogLevel crit
```

– инструкция для журналирования сообщений уровня crit и выше. Список доступных уровней идентичен списку *syslogd*. Будьте осторожны при журналировании сервера предприятия: слишком подробное описание моментально заполнит ваш диск!

Apache пишет строку передачи при каждом запросе страницы браузером. Журнал передач создается только тогда, когда вы явно запрашиваете это в файле настройки с помощью директивы TransferLog:

```
TransferLog /var/log/Apache/transferlog
```

На самом деле, *Apache* часто настраивается на ведение нескольких журналов передачи (по одному на виртуальный узел), но мы здесь ограничимся ситуациями попроще. Можно также направить ваш журнал в другую программу. Чаще всего это программа *Rotatelogs*, она периодически закрывает журнал и начинает новый. (*Rotatelogs* включена в состав *Apache*. Не путайте ее с утилитой более общего назначения *logrotate*, которой мы займемся в следующий раз.) Директива, позволяющая передать журнал в вашу программу через канал, выглядит так:

```
TransferLog "| rotatelogs"
```

Apache пишет журналы передач в формате, известном как «общий формат журналов» (common log format). Он широко поддерживается большинством web-серверов, инструментами анализа файлов журналов и рассмотрен во врезке *«Разбор общего формата журналов»*, выше.

Еще один журнал, иногда оказывающийся полезным для диагностики – журнал *X*-сервера, обычно **/var/log/Xorg.0.log**. Этот файл переписывается каждый раз при перезапуске X-сервера. 0 в имени файла – это номер дисплея *X*-сервера, и если у вас в системе несколько мониторов, вы можете найти дополнительные файлы, относящиеся к разным дисплеям.

Наконец, существует собственный «поток сознания» ядра – сообщения, которые оно создает при загрузке. Они даже не записываются в файл (многие из них генерируются на ранней стадии процесса загрузки, до того, как станет доступна файловая система), а хранятся в памяти ядра – «кольцевом буфере [ring buffer]», который отображает команда *dmesg*. Некоторые дистрибутивы скидывают вывод *dmesg* в файл на поздних стадиях загрузки системы; например, Red Hat и Fedora пишут его в **/var/log/dmesg**. Большая часть этих сообщений создается модулями ядра при попытке определения и инициализации ассоциированного с ними оборудования, и они загадочны даже по стандартам Linux. Их внимательное изучение при случае поможет определить, распознается ли ваше оборудование, но для большей части сообщений *dmesg* есть два выхода: игнорировать их или отправить гуру для удаленной диагностики.

В следующий раз мы рассмотрим некоторые типичные журналы; а пока –

```
LXF91 14:41:54 from chris: конец руководства
LXF91 14:42:44 from chris: последнее сообщение повторилось 42 раза LXF
```

› Интервью с прародителем Syslog, Эриком Оллменом, опубликовано на стр. 28.

Разбор общего формата журналов

› Существует также «комбинированный» формат журнала, в нем больше информации.

Syslog.conf парой слов

› Файл настройки определяет, что произойдет с каждым сообщением, посланным в syslog.

» **Через месяц** На втором уроке мы научимся собирать, анализировать и обновлять файлы журналов.

Neo1973 – начало мобильной революции

Генеральной целью проекта OpenMoko является превращение телефонов из стремительно дешевеющих игрушек в мобильные компьютеры с открытой платформой и возможностью пользоваться всеми благами, которые предоставляют системы GNU/Linux на рабочих станциях. За развитием проекта внимательно следит **Денис Шмаков**.

Апрель 1973 года. Доктор Мартин Купер совершает первый звонок с первого в мире мобильного телефона. И сообщает своему конкуренту из Bell Labs, что тот проиграл. **Neo1973** означает 'Новый 1973-й'. Спустя 34 года команда OpenMoko собирается совершить новую революцию в коммуникациях, совместив открытую программную платформу на основе GNU/Linux и открытую аппаратную часть от First International Computer. GSM-совместимый смартфон Neo1973 станет первым устройством, работающим под управлением OpenMoko.

Что же представляет из себя этот дистрибутив? Ядро Linux 2.6.17.14, X.Org 7.1, среда GTK+ 2.6.10, а также несколько новых библиотек: *libmokocore* (реализует некоторые специфические для смартфона функции управления), *libmokoui* (содержит набор удобных элементов пользовательского интерфейса на базе GTK+), *libmokonet* (предоставляет высокоуровневые функции GSM и GPS), *libmokopim* (позволяет легко управлять персональными данными пользователя). Среди предустановленных приложений будут программа для набора номера, менеджер контактов, календарь, медиаплеер, браузер и некоторые другие. Однако, поскольку OpenMoko базируется на достижениях существующего проекта OpenEmbedded, тысячи дополнительных приложений доступны для установки уже сейчас. Простой и удобный менеджер пакетов *ipkg* позволит легко находить и устанавливать новые программы. «Приложения - это рингтоны будущего», – говорит Шон Мосс-Пулц [Sean Moss-Pultz], один из основателей и главный идеолог проекта OpenMoko.

Графический интерфейс OpenMoko представляет собой симбиоз компонентов среды GPE (мини-GNOME), оконного менеджера Matchbox и базовых приложений на основе libmokoui. Предельно простой и удобный, оконный менеджер Matchbox тем не менее обладает довольно высокой настраиваемостью и масштабируемостью. Он уже используется во многих существующих PDA, webкиосках и других портативных устройствах. Помимо GPE, можно использовать основанную на Qt среду OPIE (миниатюрный аналог KDE), однако основная масса разработчиков не желает включать её в официальный дистрибутив для сохранения консистентности (*X11 + GTK + libmokoui*). Впрочем, не стоит забывать о том, что OpenMoko является лишь первопроходцем в данной области. Вполне вероятно, что вслед за ним появятся и другие дистрибутивы для открытых смартфонов серии **Neo1973**, которые могут взять за основу формулу *X11 + Qt + OPIE*.

Neo1973 поддерживает сетевое соединение через USB и может быть использован как GPRS-модем/брандмауэр/NAT-маршрутизатор для компьютера. Для ядра Linux разработан патч, делающий возможным многопоточный доступ к функциям GSM, поэтому пользователь сможет одновременно совершать звонки и бродить по Интернету.

Помимо сетевого USB-соединения возможно подключение в роли хоста, что позволяет смартфону работать с другими USB-устройствами (видеокамерами, мобильными телефонами, плеерами) подобно персональному компьютеру, при условии наличия у этих устройств собственного источника питания. В **Neo1973** не предусмотрена поддержка Wi-Fi, но самые нетерпеливые могут воспользоваться внешним адаптером, просто подключив его к USB-порту.

Весь дистрибутив занимает 64 Мб, оставляя равный объём памяти для дополнительных приложений и данных. Увеличение памяти до 2 Гб возможно за счёт внешних карт microSD.

Хотя **Neo1973** является пока что единственной целевой серией для OpenMoko, некоторые энтузиасты уже пытаются перенести его на другие, закрытые платформы. Это объясняется отсутствием у текущей модели **Neo1973** некоторых устройств, например адаптера Wi-Fi и видеокамеры. Однако, по мнению многих, подобные попытки могут повредить сотрудничеству с FIC. «Зачем тратить время, занимаясь инженерным анализом, если теперь у нас есть производитель оборудования, с которым мы можем общаться напрямую?» – говорят они. Будущие спецификации **Neo1973** во многом определяются пожеланиями (потенциальных) пользователей, а не корпоративным планированием FIC. Пожалуй, налаживание открытого диалога сообщества с производителем является не менее важным событием, чем непосредственно разработка дистрибутива. Примечательным является и тот факт, что FIC не только прислушивается к пожеланиям разработчиков и энтузиастов OpenMoko, но и оплачивает труд многих из них. Вне всякого сомнения, начальный успех этой новой мобильной революции зависит от ее взаимного сотрудничества с OpenMoko.

OpenMoko призван быть дружественным не только для конечных пользователей, но и для разработчиков. Превосходный (хотя и неофициальный) инструмент *MokoMakefile*

позволяет получить настроенную и готовую к использованию среду разработки буквально несколькими командами. *MokoMakefile* – это обычный *Makefile*, который настраивает структуру каталогов, скачивает последние версии всех необходимых файлов, производит компиляцию, установку и «подгонку» пакетов, и даже синхронизирует настройки с другими разработчиками. В результате компиляции получается полностью готовая к работе SDK и образ файловой системы в виде tar-архива, который затем можно запустить в эмуляторе QEMU. Среда разработки занимает примерно 7 Гб, а целевой образ компилируется на среднем персональном компьютере около полудня. Подробные инструкции по установке SDK можно найти на сайте OpenMoko Wiki (**wiki.openmoko.org**). Там же, в подразделе 'Sections by target audience' можно получить детальную информацию о стратегиях и инструментах для разработки приложений/системы и отладки аппаратного обеспечения OpenMoko/Neo1973.

Готовые образы файловой системы с OpenMoko доступны на **buildhost-old.openmoko.org** в каталогах **/releases/gta*/tmp/deploy/images** (единым файлом), **/releases/gta*/tmp/deploy/ipk** (в виде установочных пакетов *ipkg*), **/releases/gta*/tmp/rootfs** (развернутая файловая система).

К тому моменту, когда вы будете читать эти строки, **Neo1973** уже должен быть доступен для заказа в следующей базовой комплектации:

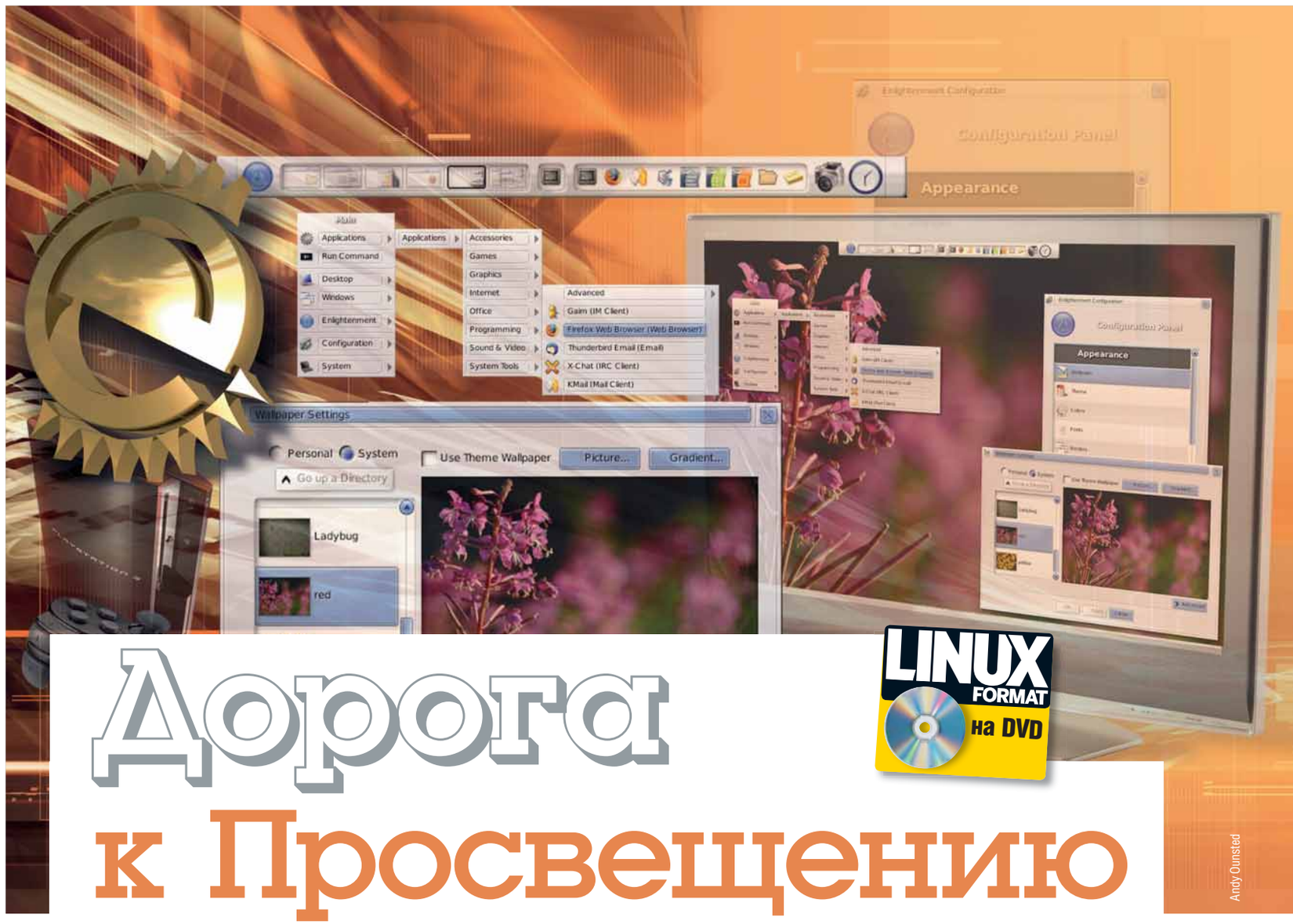
- смартфон **Neo1973**;
- батарея;
- наушники и микрофон;
- зарядное устройство;
- чехол;
- стилус;
- карта памяти microSD 512 Мб;
- USB-кабель
- инструкция по использованию и гарантия.

Стоимость смартфона будет составлять около $350. Уведомление о начале продаж можно получить, подписавшись на новостную рассылку. По заявлению OpenMoko, после начала продаж на **openmoko.com** появится форма online-заказа, а пока что можно добавить себя в списки потенциальных покупателей здесь: **wiki.openmoko.org/wiki/Buying_Interest_List** Массовые продажи потребительской версии намечены на сентябрь. LXF

Спецификации смартфона Neo1973

ПРОИЗВОДИТЕЛЬНОСТЬ		
	Процессор	Samsung s3c2410 SoC до 266 MHz
	Память	128 MB SDRAM
	Постоянная память	128 MB NAND Flash и слот расширения для microSD
СВЯЗЬ		
	GPRS	Texas Instruments GPRS (2.5G, без EDGE)
	Bluetooth	Delta DFBM-CS320 Class2
	GPS	Global Locate Assisted GPS chip
	USB	1.1, незапитанный
ЭРГОНОМИКА		
	Размеры	120.7x62x18.5 мм
	Вес	185 г
	Экран	2.8" VGA (480x640) TFT, повышенной чёткости
	Ввод	Сенсорный экран + стилус или внешняя Bluetooth-клавиатура
	Кнопки	"Питание" и "Экстренный вызов"
	Аудио	2,5мм аудио разъём
	Батарея	1200 mAh (зарядка через USB), совместима с Nokia BL5C

Как программное, так и аппаратное обеспечение смартфона постоянно совершенствуется, поэтому к моменту начала массовых продаж потребительской версии возможны некоторые изменения конфигурации, например, добавление видеокамеры.



Дорога к Просвещению

LINUX FORMAT на DVD

Andy Ounsted

Оконный менеджер *E17* появился на… *PlayStation 3*! **Натан Сандерс** пояснит, как проблемы Yellow Dog привели к этому проекту.

Тем, кто азартно пытался запустить Linux на PlayStation 2 или принял вызов Майкла Робертсона [Michael Robertson], внедряя Linux на Xbox, не повезло. А если вы просто хотели заставить свои устройства работать как полагается, поздравляем. Sony взяла на себя всю черную работу по установке Linux на PS3 – игровую приставку нового поколения. Но если вы думаете увидеть знакомый GNU/Linux, вас ждет сюрприз.

Дело в том, что фирма Terra Soft для своей PS3-версии дистрибутива Yellow Dog вытащила из застоя оконный менеджер *Enlightenment* [по-англ. так называется эпоха Просвещения, а заодно и «облегчение», – прим. ред.], и после пяти лет разработки *Enlightenment 17 (E17)* готов к игровым подвигам. А началось все в 2003 г., когда Sony выбрала для своей новой приставки процессор Cell (специализированная производная от архитектуры PPC). Поясним…

Пара слов о Cell

Cell, процессор PS3, содержит основной модуль, совместимый с другими устройствами архитектуры PowerPC на уровне двоичного кода, и восемь специализированных блоков – грамотная программа обязана учитывать эти особенности. Как и все многоядерные процессоры, Cell не особо эффективен для web-серфинга или обработки текста, но дает огромный выигрыш при возможности распараллеливания, например, при кодировании аудио или рендеринге сцен 3D-графики. Yellow Dog 5 поставляется с библиотекой разработчика Cell SDK от Barcelona Computing Centre. Кай Стаатс, руководитель Yellow Dog, назвал ее «хорошей отправной точкой» для оптимизации программ в Cell; он также рекомендует компилятор от RapidMind (www.rapidmind.net) и библиотеки MCF Cell (www.mc.com).

При чем тут Yellow Dog

Дистрибутив Yellow Dog появился в 1999 году и был ориентирован на энтузиастов Linux, имеющих в своем распоряжении Mac. Чтобы занять эту нишу, Terra Soft разработала ОС для архитектуры PowerPC/Apple, хотя практически всюду использовались чипы x86 от Intel. Terra Soft продолжает продавать свой дистрибутив на базе KDE владельцам Mac'ов, а также занимается спецприложениями для суперкомпьютерных расчетов.

Но в 2005 г. Apple объявила о переходе на платформу Intel, и Terra Soft пришлось пересмотреть стратегию: улучшив свою суперкомпьютерную ОС Y-HPC, пригодную и для PowerPC, и для x86, а также разработав узкоспециализированное ПО для анализа генных цепочек, фирма стала главным игроком на поле суперкомпьютеров PPC, так что Terra Soft отнюдь не бедствовала, когда Sony выбрала PPC за основу PS3.

Версия Yellow Dog для PlayStation3 нуждалась в компактном и эстетичном рабочем столе, и летом 2006 г. Terra Soft обратилась к ведущему разработчику *Enlightenment* Карстену Хайцлеру [Carsten Haitzler] за долгожданным, но сильно запаздывающим E17. Хайцлер хоть и обрадовался, но из осторожности отказался квалифицировать версию даже как бета, пока все ошибки в *Enlightenment* не будут исправлены: по его мнению, «версия альфа – это когда функциональность отработана, но осталось несколько известных ошибок».

Yellow Dog 5: с суперкомпьютера на PlayStation

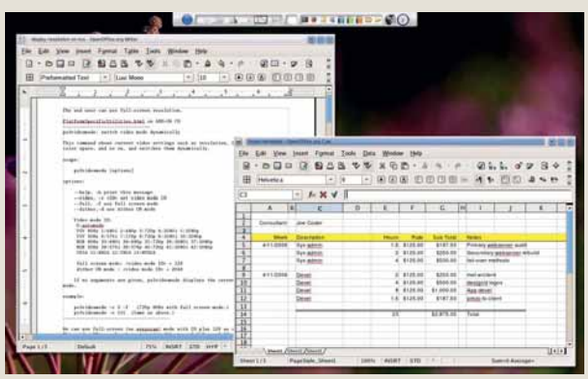
Перенос Yellow Dog на PS3 был сравнительно простой задачей. Terra Soft стала производителем первого коммерческого дистрибутива с поддержкой Cell, разработав его еще в 2005 г., в сотрудничестве с Mercury Computers. Кроме того, если отвлечься от архитектуры процессора, PS3 – тот же персональный компьютер: жесткий диск, 6 портов USB 2.0, слоты для флэш-карт, Ethernet, Wi-Fi и привод для чтения оптических дисков.

Однако совершенства Yellow Dog 5 в PS3 пока не достиг. Разрешение экрана телевизора гораздо меньше, чем разрешение любого компьютерного монитора, а значит, доступное пространство на телевизионном экране существенно ограничено. Поэтому инсталляция Yellow Dog 5 на PS3 происходит в текстовом режиме и требует последующей настройки. На момент написания статьи были также проблемы с поддержкой Wi-Fi карт, хотя Кай Стаатс рассчитывает в этом вопросе на содействие Sony.

Не все гладко и с устройствами Bluetooth. Данные с Blu-ray дисков Yellow Dog на PS3 читать умеет, но по юридическим причинам в поставку дистрибутива нельзя включить кодеки для проигрывания Blu-ray видео. На вопрос, нельзя ли убедить Sony снять ограничения на чип RSX, из-за которых Yellow Dog не может использовать графический ускоритель, Стаатс только и мог ответить: «Да говорили мы им»…

Yellow Dog – не единственный дистрибутив для PS3. На его территорию могут покушаться Fedora и Gentoo, а в будущем – и другие дистрибутивы с поддержкой PPC. Сейчас, однако, рядом с PlayStation 3 на магазинных полках соседствует один Yellow Dog, и только он обладает статусом официальной поддержки оборудования PS3. Насчет его спецификаций – YDL5 основан на Fedora Core 5 (ядро Linux 2.6.16) и включает стандартные средства, например, Yum, офисный пакет *OpenOffice 2.0*, компилятор GCC 4, среды KDE 3.5 и Gnome 2.14 и браузер *Firefox 1.5*.

Дистрибутив можно заказать на DVD или (тут количество ограничено) уже предустановленным на PS3, через сайт www.terrasoftsolutions.com, за $49.95. Там же принимаются заказы на игровую приставку PlayStation 3 с предустановленным Yellow Dog 5. Общедоступные для загрузки версии должны появиться к январю. Хотя YDL5 теоретически пригоден и для оборудования PPC от Apple, Terra Soft предупреждает: тестирования пока не проводилось.



Новый Yellow Dog 5 дарит вашим любимым приложениям новую консоль…

Сдав продукт в ноябре 2006 года, он назвал его *Enlightenment 17* пре-альфа.

Понятно, почему Terra Soft ухватилась за *Enlightenment*, несмотря на его незрелый статус: по словам Хайцлера, эта оконная среда придает Yellow Dog «сексапильный глянец».

Также понятно, что внешность – не главное: в отличие от KDE и Gnome, *Enlightenment* славен быстрой загрузкой (буквально за несколько секунд); он быстр и не требователен к оперативной памяти. Несмотря на мощный процессор, PS3 имеет всего 256 МБ ОЗУ – а Yellow Dog 5 с *Enlightenment* обходится 65 МБ, оставляя место для других приложений.

> «По словам Хайцлера, Enlightenment придает «сексапильный глянец.»

Версия E17 обещала гигантский скачок (предыдущий релиз DR16.0 вышел еще в 2000 г.), но создавалась она совершенно заново, с нуля. Потенциальные потребители, соблазненные было классным графическим интерфейсом, уживающимся даже на самых медленных системах, помаленьку разбегались: разработка чересчур затянулась. Генеральный директор Terra Soft Кай Стаатс [Kai Staats] понимал это, и решился помочь проекту. Чтобы придать ускорение команде Хайцлера, Terra Soft нацелила ее на несколько конкретных задач, которые обе стороны расценили как неотложные и необходимые, и по особо важным пунктам назначила денежные премии для программистов.

Стаатс победно сообщает, что все главные задачи завершены, включая полную переделку кода рабочего стола и управления фоновым изображением, введение нового селектора файлов 'fm2', поддержку drag-and-drop и принятие стандарта Freedesktop.org для рабочего стола, меню и иконок.

Все это рано или поздно произошло бы и без Terra Soft, но спонсорская помощь существенно ускорила процесс. Стаатс уклонился от озвучивания премиальных сумм, но судя по объему проделанной работы, программисты не за морковкой гнались. Причем продукт, созданный при поддержке Terra Soft, выходит под той же лицензией BSD, что и весь *Enlightenment*.

Не только у *Enlightenment* есть красивый рабочий стол с анимацией и эффектами: KDE и Gnome тоже такое предлагают. Но не на PS3. Оконные менеджеры *Compiz* or *Beryl* построены на *Xgl* and *AIGLX*, то есть на аппаратном ускорении OpenGL, применение которого в оборудовании Sony для PS3 ограничено. *Enlightenment* же в OpenGL не нуждается.

Получше Cairo?

Шикарная 2D-анимация *Enlightenment* обязана своим появлением графической библиотеке *Evas*; кому-то захочется сравнить ее с аналогичной библиотекой Cairo, популярной в сообществе Gnome. Хайцлер предупреждает, что подобное сравнение некорректно. Cairo – небольшая библиотека, по мере возможностей повышающая эффективность вывода векторной графики с помощью аппаратного ускорения, а *Evas* гораздо богаче: здесь разработчики могут работать с графическими примитивами как с объектами, легко ими управлять и анимировать их. По мнению Хайцлера, «когда *Evas* использует Cairo как движок рендеринга вместо собственного движка, скорость обработки снижается вдесятеро. Другое дело, что *Evas* ориентирована скорее на растровую графику, а Cairo – на векторную». *Evas* может использовать и OpenGL, но встроенный движок *Evas* справляется с задачей гораздо лучше. »

Виджеты в Enlightenment 17

Как и в Mac OS X, KDE и Gnome, в *Enlightenment 17* могут использоваться апплеты различного назначения: мониторинг электронной почты, просмотр изображений, блокнот и т.д.

Для работы с апплетами используется *Gadcon* (Gadget Container): сами апплеты сложены на «полку» (панель рабочего стола), а вызываются из панели управления. В будущих версиях *Enlightenment* разработчики надеются обеспечить встраивание апплетов в меню и рамки ок

» Некоторые из 2D-эффектов *Enlightenment* реализованы и в Gnome, и в KDE (через *Qt* и *GTK*), но отнюдь не повсеместно. Например, в оформлении некоторых тем KDE используются анимированные кнопки и заголовки окон. Иногда по скорости работы и потреблению памяти библиотеки *Enlightenment Foundation Libraries* (*EFL*) отличаются от базовых библиотек прочих оконных сред, за счет своей элегантности, но это не техническое расхождение, а философское: у KDE упор на функциональность, у Gnome – на простоту использования, а у *Enlightenment* – на яркие эффекты.

Прочный фундамент

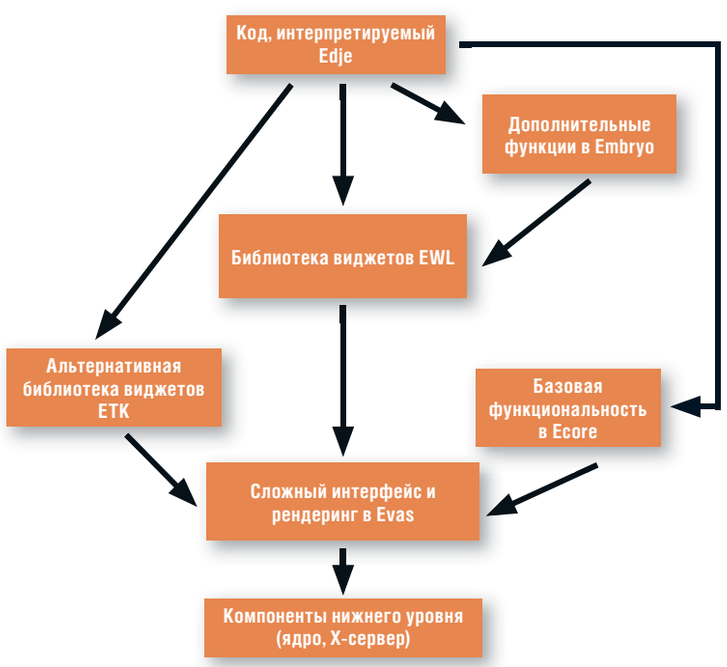
Оконный менеджер *Enlightenment* – лишь одно из многих приложений на базе *EFL*, откуда взяты компоненты для E17, точно так же, как KDE использует графическую библиотеку Qt, а Gnome – GTK. У *EFL*, однако, несколько иная функциональность. Фактически это целый набор библиотек, с широким охватом. Команда *Enlightenment* оправдывает свою медлительность необходимостью совершенствовать эти библиотеки. Документацию, информацию и даже примеры кода, показывающие, как удобно программировать с *EFL*, можно найти на http://enlightenment.org. А сейчас вкратце рассмотрим основные библиотеки в составе *EFL*:

» **Imlib2** – библиотека для низкоуровневой обработки изображений и создания анимации.

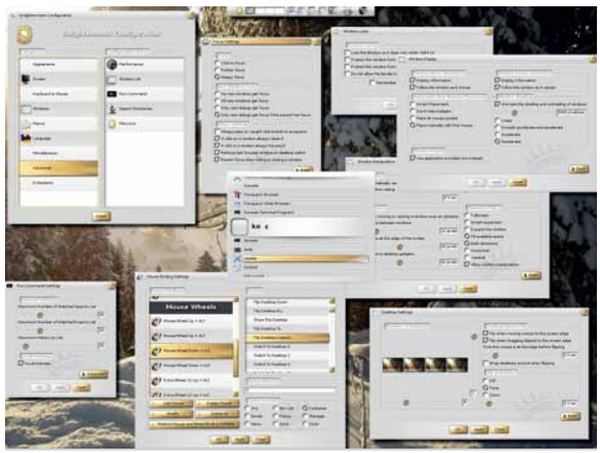
» **EPEG** – библиотека масштабирования изображений в формате JPEG, полезная для файловых менеджеров или приложений для просмотра фотографий (создание миниатюр). Обе библиотеки по скорости работы превосходят аналоги из Gnome и KDE (*gdk-pixbuf* и *QImage*).

» **Epsilon** – то же, что EPEG, но работает с большинством графических форматов; по возможности, использует *Imlib2* и EPEG для повышения быстродействия Совместима со стандартами Freedesktop.org, так что уменьшенные копии изображений будут храниться в папке **~/.thumbnails/** в формате PNG, и другим совместимым приложениям не придется дублировать работу по масштабированию.

Основные библиотеки EFL



Код, поступающий из архива *EET*, обрабатывается либо *Edje*, либо *Embryo*. *EWL* и *ETK* поставляют интерфейсные виджеты, а *Ecore* – функциональность остальных библиотек (по соображениям модульности, она помещена в отдельную библиотеку). Evas осуществляет графический рендеринг и выдает результат обработки X-серверу.



› Положение окон можно фиксировать, виртуальные рабочие столы можно переключать, тронув курсором край экрана: *E17* – мечта любителя поиграть с настройками!

» **EVFS** – библиотека для работы с файловой системой. Аналогична библиотекам KIO и GnomeVFS и предоставляет набор протоколов для работы с файлами, включая поддержку bzip2, FTP, gzip, Samba, SFTP, и tar.

» **Ecore** объединяет несколько маленьких библиотек в один пакет, ради простоты и модульности.

» **Edje** – библиотека для работы с интерфейсом приложений; ее забота – организовать. интерфейс по максимуму удобства для пользователя. Индивидуальная настройка интерфейса каждого приложения сохраняется в EDC (*Edje* Data Collection) в файле .edj, с описанием расположения элементов и прочей сопутствующей информацией..

» **EET** – библиотека для архивирования данных EDC. Обеспечивает высокую скорость доступа к данным. Степень сжатия в данном случае не главная забота: важнее, чтобы пользователь не ощутил задержки при загрузке элемента интерфейса. При наличии готового EDC построить интерфейс можно несколькими API-вызовами из *Edje*.

По теме Edje

Как и многие библиотеки *EFL*, *Edje* призвана облегчить труд разработчиков приложений и тем. Уровень абстракции, обеспечиваемый библиотекой, позволяет разработчикам создавать необычные интерфейсы приложений, при этом достаточно хорошо интегрированные с *Enlightenment*. Подробное руководство пользователя и «книга рецептов» к данному продукту – на www.enlightenment.org/libraries/Documentation и http://edevelop.org.

Имена библиотек легко расшифровать: виджеты реализуются библиотекой *EWL* (*Enlightenment Widget Library*); альтернативная библиотека виджетов *ETK* предоставляет GTK-подобный API. Виджеты обрабатываются в *Edje*, а на экран выводятся через *Evas* (*Enlightenment canvas*, холст): сложные интерфейсы приложений *Enlightenment* преобразуются в команды X-сервера, позволяя разработчику не думать о конкретной архитектуре, разрешении экрана, глубине цвета и других параметрах. *Evas* даже умеет преобразовывать графические форматы и выполнять сглаживание (anti-aliasing).

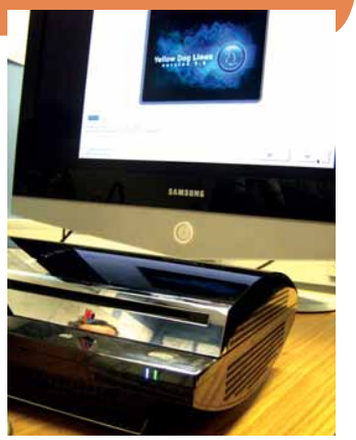
Интересно, что *Evas* способна выводить на экран и не только методами X-сервера. Это, да еще низкие требования к производительности, делает *Enlightenment* пригодным для КПК, мобильных телефонов и т.п. (разработчик *Enlightenment* Дэвид Сейкел [David Seikel] тестировал рабочий стол на компьютере с процессором Pentium с тактовой частотой 100 МГц, чтобы удостовериться в его эффективности).

При всех преимуществах *EFL*, прикладных программ здесь позорно мало. Дальше менеджера фотоальбомов *EPhoto*, файлового менеджера на основе evfs *Entropy*, музыкального проигрывателя в стиле *iTunes Emphasis* с движком MPD Music Player Daemon и программы для извлечения звука из аудио-CD *Extrackt* (с поддержкой

Установка Linux на PS3

Установка Linux на PS3 довольно проста, но требует некоторой подготовки. Понадобится установочный DVD, клавиатура и мышь с USB-интерфейсом и устройство хранения данных. Включите приставку, затем выберите в меню Настройки (Settings) > Настройки системы (System Settings) > Форматирование (Format Utility) > Форматировать жесткий диск (Format Hard Disk) в ОС Sony Game. Нажмите «Да» («Yes») в появившихся окошках, а затем «Подробнее» («Custom»). Тут нужно задать размер системного раздела. Terra Soft рекомендует 10 ГБ – более чем достаточно, т.к. Yellow Dog 5 с оконными менеджерами E17 и Gnome умещается примерно в 3 ГБ.

На следующем этапе установки потребуется загрузчик. Скачайте установщик загрузчика с сайта **www.playstation.com/ps3-openplatform** и сохраните его как **/ps3/otheros/otheros.self**, потом скачайте сам загрузчик с **www.terrasoftsolutions.com/support/install/ps3/otheros.bld** и сохраните как **/ps3/otheros/otheros.bld**. Затем перепишите установщик загрузчика и загрузчик на USB-брелок или другой носитель, распознаваемый PS3. Перезагрузите систему при вставленном USB-брелке и выберите в меню пункт Настройки (Settings) > Настройки системы (System Settings) > Установить другую ОС (Install Other OS). Установите загрузчик, следуя появляющимся на экране инструкциям. После этого в меню выберите пункт Настройки (Settings) > Настройки системы (System Settings) > ОС по умолчанию (Default System) и выберите «Другая ОС» (Other OS). Теперь все готово для установки Yellow Dog 5. Sony GameOS все равно сохранится, для ее загрузки можно в любой момент набрать boot-game-os в командной строке загрузчика. Загрузите приставку и вставьте в привод установочный DVD Yellow Dog 5. Выберите Install для запуска программы установки *Anaconda*, как в Fedora или Red Hat. После этого вами будет руководить приятный графический гид, подробно объясняющий все непонятные для начинающих моменты.



› **Сыграем?**

Дополнительная информация – на **www.terrasoftsolutions.com/support/installation**.

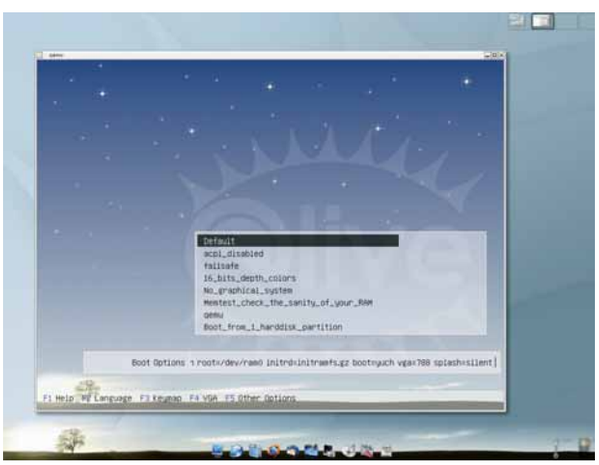
OGG, MP3 и WAV) дело не идет. (Все они находятся на **www.get-e.org**.) Yellow Dog должен позаботиться о привлечении разработчиков на платформу *Enlightenment*, а пока пользователи пробавляются ПО, разработанным для KDE, Gnome и других оконных сред.

Если вы, накаляясь, дожидались *E17* пять долгих лет, вам скоро полегчает. Разработка *E17* заметно активизировалась, от чего выиграл весь релиз. Сейкел говорит: «*E17* создавался с нуля: *Enlightenment* не боится начинать с нуля, если это оправдано. Но *E18* строится на солидной базе *E17*, и дело должно пойти гораздо быстрее. Правда, мы кое-что переделаем, так что ожидается некоторый прогресс. Я думаю, этот релиз выйдет гораздо быстрее, чем *E17*».

Стаатс не закрывает глаза на опасность использования передового, но незрелого рабочего стола при наличии стабильных альтернатив: для приверженцев старых Linux-традиций на DVD Yellow Dog 5 предусмотрены и Gnome, и KDE, и соответствующие репозитории. Но он утверждает, что пара *E17* + YD5 вполне готова к работе. Terra Soft с лета 2006 г. добросовестно тестирует дистрибутив на PlayStation 3, рабочих станциях и даже серверных платформах.

Больше чем игра

Союз Terra Soft и *Enlightenment* ждет яркое будущее. Стаатс надеется, что использование Yellow Dog на PlayStation 3 расширит его аудиторию. «Игра притянет людей, а в придачу они получат дистрибутив, который их удержит», говорит он. «Благодаря использованию Yellow Dog Linux на PlayStation 3 дистрибутив разойдется по всему миру. Опять же, увеличится популярность Linux, и рядовые пользователи поймут, что PS3 – нечто большее, чем обычная игровая приставка.» Игроки, использующие *OpenOffice.org*? Ну, если их *просветить*… LXF



› **Live-дистрибутив *Elive*, по умолчанию использует *Enlightenment*; здесь вы его видите в эмуляторе Qemu.**

ОТКРОВЕНИЯ GIMP

GIMP пробудился. **Ник Вейч** выясняет, как его оживляют в преддверии версии 2.4, почему проект GEGL так разросся, и что пользователи делают с нашим лучшим графическим инструментом.

Paul Blachford

Попросите людей назвать какую-нибудь открытую программу; скорее всего вы услышите *OpenOffice.org*, *Firefox*, возможно, Apache – в зависимости от того, с кем говорите. Вспомнит ли кто-то про *GIMP*? Как ни странно, он не приходит людям на ум первым. Подобно гравитации, *GIMP* присутствует незримо. Как и скрипты инициализации, он, пожалуй – нечто встроенное в ваш дистрибутив и воспринимаемое как данность.

Не думайте, что *GIMP* не популярен. Он вполне способен удерживать свои позиции в списке 20 лучших открытых проектов по количеству обращений на SourceForge. Поскольку это редактор изображений, устанавливаемый по умолчанию во всех настольных дистрибутивах Linux, трудно представить истинного линуксоида, который время от времени не прибегал бы к его услугам – в основном потому, что *GIMP* очень гибок по части обработки графики. Порты для Mac и Windows значительно расширили аудиторию *GIMP*, а с учетом дефицита бесплатных программ обработки графики для этих платформ, не удивительно, что его принимают на ура.

Но и у лучшей графической программы для Linux есть свои проблемы. Немало воды утекло с момента последнего крупного релиза (версия 2.2 вышла в декабре 2004 г.), и хотя добавлялись новые функции (и грядет релиз 2.4), по-прежнему налицо довольно большой список невыполненных пожеланий.

Предыстория

Как и другие социально значимые проекты, *GIMP* был впервые представлен широкой публике в группе Usenet-новостей comp.os.linux.misc. В 1995 г. два студента из Беркли, Спенсер Кимболл [Spencer Kimball] и Питер Мэттис [Peter Mattis], попытались создать для фанатов свободного ПО нечто взамен *Photoshop*, который полюбился художникам на платформах Windows и Mac. Их ранний код был, прямо скажем, лоскутный и довольно нестабильный, и использовал откровенно ужасный графический интерфейс *Motif*. И все же это был прекрасный фундамент для отличного графического инструмента: 24-битный цвет, поддержка целого ряда файловых форматов, разумная система расширений и функция отката.

GNU Image Manipulation Program мгновенно стала гвоздем сезона.

ЗАНОВО

Искусство GIMP: Джош

Одна из лучших черт *GIMP* – изумительная гибкость, позволяющая достичь практически любого эффекта, поддающегося описанию. Вместе интерфейсами для скриптовых языков, возможности для расширения этого ПО вдохновили многих разработчиков и пользователей на создание подлинно удивительных новых способов превращать сборища точек в эффектные изображения.

Одно из наиболее гибких расширений – MathMap. Оно позволяет пользователям создавать вычисленные (процедурные) изображения, основанные на данных о пикселях, рассчитываемых по формулам, и поставляется с десятками предустановленных примеров. Впрочем, Джош Соммерс [Josh Sommers] – не из тех, кто просто применяет эти преднастройки: показанный здесь фантастический рисунок является результатом сложного кодирования наряду с художественными усилиями.

Соммерс рассказывает нам: «Больше всего мне нравится в *GIMP* то, что каждый рисунок размещается в собственном окне со своими органами управления, почти как в многоэкземплярном приложении. Когда я применяю фильтр, требующий много времени на прорисовку, я могу перейти к другому изображению и работать с ним, пока тот эффект не выполнится. Применяя фильтр в Photoshop, я должен ждать завершения его работы, и вдохновение улетучивается».

Другие работы Джоша, а также прекрасное руководство по созданию этого эффекта вы найдете на его Flickr-страничке: **www.flickr.com/photos/joshsommers/378056770**.

Двум студентам неожиданно пришлось разбирать запросы все возрастающего числа пользователей, но они были упорны. Попутно Мэттис решил, что *Motif* действительно ужасен, и родился целый новый проект: *GTK*. Он затевался всего лишь как инструментарий для *GIMP*, но хорошие вещи обычно идут нарасхват везде.

В конечном итоге Кимболл и Мэттис окончили колледж и нашли настоящую работу, но *GIMP* не умер – лицензирование под GPL означало, что он мог быть продолжен. Вот он и продолжался, с различными командами разработчиков, сквозь взлеты и падения, несколько лет подряд. (Если вам интересна подробная история этого приложения, очевидно, лучше всего поискать ее на собственном сайте *GIMP*, **www.gimp.org**, раздел About.)

Но, видимо, *GIMP* был уж слишком хорош. Его ранний успех и весьма открытая разработка означали, что желающие создавать графическую программу неизбежно попадали в *GIMP*. Здесь не было ни великих расколов проекта, ни конкуренции за его пределами, ни выбора для конечных пользователей. Выбор – основа философии свободного и открытого ПО, но здесь он отсутствовал: не потому что хакеры не были способны соперничать, а потому, что предпочитали этого не делать. Из-за недостатка конкуренции и отсутствия спонсорской поддержки корпораций (*см. врезку справа, «Кто платит за GIMP?»*) набор функций *GIMP*, который раньше, в бурные дни конца девяностых, действительно в чем-то превосходил закрытых конкурентов типа *Photoshop*, начал отставать от них.

Серия *GIMP 2.x* в известной степени продемонстрировала возврат былой формы. Хотя темп разработки вряд ли по душе рьяным пользователям, по крайней мере, стали заметны связность, целеустремленность и ощущение стабильного прогресса. Промежуток между основными релизами будет укорачиваться (возможно, шесть месяцев между 2.4 и 2.6), а настоящие новшества вновь попали в поле зрения. После появления *GIMP 2.4* где-то в этом году определенно имеет смысл обновиться. »

Кто платит за GIMP?

Если коротко, то никто. *GIMP*, конечно, свободное (free) ПО, и в данном случае это означает и бесплатность (free as in beer), и свободу (free as in speech). Но большинство солидных проектов финансируются, хотя и косвенно. *OpenOffice.org*? Несомненно, бесплатен для использования, но разработка на самом деле оплачивается рядом компаний, включая Sun и Novell, которые специально нанимают под него сотрудников. Python? Google берет на себя большую часть стоимости разработки этого языка, оплачивая работу Гвидо ван Россума [Guido van Rossum] и многих других разработчиков Python. Само ядро Linux? Это, вероятно, самый спонсируемый проект – все основные поставщики дистрибутивов оплачивают работы над ключевыми аспектами ядра.

Но *GIMP*, один из наиболее успешных представителей настольных программ для Linux, разрабатывается без спонсорской помощи. Основная команда разработчиков очень невелика, и их сторонние обязательства ограничивают темпы развития. Страдают также документация и web-сайт. Итак, Google, Novell, HP, IBM и Ко! Если вы читаете это, то знайте: есть некий сектор среди предметов рабочего стола первой необходимости, которому не помешает ваша забота. Обычные пользователи тоже могут помочь – см. врезку в конце этой статьи.

Что нового в 2.4

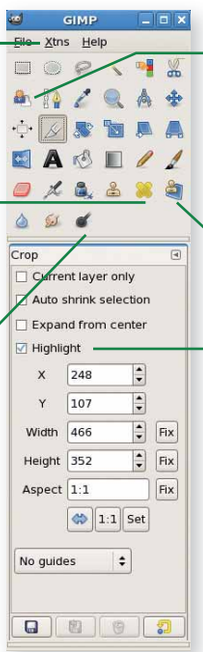
Раскурочьте ваши пиксели новыми инструментами.

Новый релиз предусмотрит обычные исправления ошибок и улучшения, в том числе ряд давно обсуждаемых вещей: например, термины, используемые в диалогах фильтров.

Также в коде заметно больше хитрых приемов. Обработка EXIF в JPEG теперь намного лучше, и *GIMP 2.4* автоматически определяет флаг 'rotate' и при загрузке изображения спросит, надо ли его повернуть. Значительно улучшен импорт векторной графики (особенно формата SVG), а также обработка текста.

Вы должны ощутить повышение производительности, поскольку сборки x86 теперь оптимизируются для конкретных процессоров. А как насчет чудесных новых функций? Читайте дальше...

Новая панель инструментов



Изменения в меню
Меню было слегка изменено, чтобы предоставить более быстрый доступ к фильтрам, не зависящим от изображения.

Целительный пластырь
Новый инструмент Helaing получил, наконец, особую иконку.

Иконки-незнакомки
Многие иконки на панели инструментов перерисованы; людям, привыкшим к старым версиям GIMP, придется не раз остановиться и подумать.

Foreground Select
Функция Foreground Selection и расположенный ниже инструмент Alignment превосходны для многослойных изображений.

Perspective Clone
Для умного клонирования.

В отрыв!
Настройки, зависимые от инструмента, теперь можно «отрывать», получая плавающее диалоговое окно.

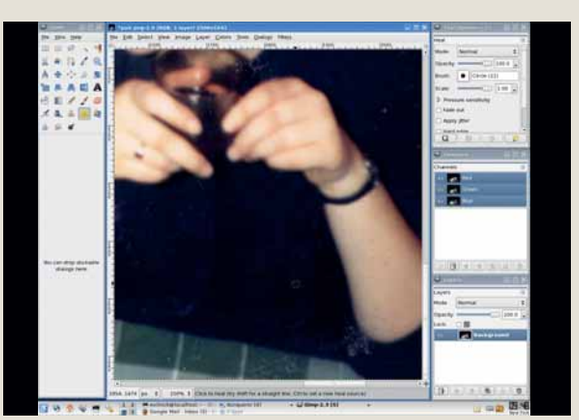
Инструмент Healing

Новшество на панели инструментов – инструмент Healing. По действию он похож на инструмент с тем же названием в *Adobe Photoshop*, и некоторое время активно комментировался в списках рассылки *GIMP*. Идея его в том, что иногда бывает нужно доработать некоторую область изображения – возможно, на объективе была пыль, поцарапан негатив или что-то еще. Инструмент Healing позволяет клонировать сходную часть изображения, потихоньку затягивая пятно – раньше это делалось обычным инструментом Clone при настраиваемых пользователем опциях. Не пытайтесь переосмыслить его, а просто пользуйтесь: это чистое волшебство.

Шаг за шагом: Используем инструмент

» Control + щелчок

Как и в случае инструмента Clone, удерживайте нажатой клавишу Control и щелкните на характерную область изображения, которую хотите использовать.



» Закрашиваем

Мазните дефект кистью, и он тут же будет исправлен – гораздо быстрее, чем при использовании инструмента Clone.

Искусство GIMP: Давид Спиньола

Давид Спиньола [David Spignola] – дизайнер из Италии. Днем он использует все дорогостоящее ПО, которое по карману его фирме, но дома для своих экспериментов в мире графики предпочитает *GIMP*: «Я использую его с релиза 2.0.1, это где-то с 2002 года, если память меня не подводит, – говорит он. – Интерфейс не особо дружественный, но к нему привыкаешь. *GIMP* почти идеален, но при очень больших файлах, с высококачественными изображениями, слегка тормозит».

Рисунок Спиньолы, показанный здесь, выполнен методом, называемым «tilt-shift miniature fake» (фальшивая миниатюра). Идея заключается в том, что ваш мозг, по-видимому, из-за просмотра слишком большого числа анимационных телешоу в детстве, ассоциирует малую глубину резкости изображения с чем-то мелким. Таким образом, чтобы обмануть мозг, малую глубину резкости можно имитировать созданием градуированных размытостей на нормальном изображении. Технически фокус нехитрый, но бывает весьма эффективен при тщательном выборе объекта.

Управление цветом

Когда о *GIMP* приходилось лишь мечтать, управление цветом пребывало во младенчестве и фактически не влияло на повседневную работу с графикой. В наши дни, когда для создания изображений все чаще используются цифровые системы и все более различаются и усложняются процессы вывода, управление цветом становится повсеместным. Попросту говоря, встроенные в изображение профили гарантируют, что цвета воспроизведутся на произвольном носителе в точности так, как было задумано. То есть изображения, которые вы видите на экране, не изменятся и при выводе на печать через PDF или любые другие форматы.

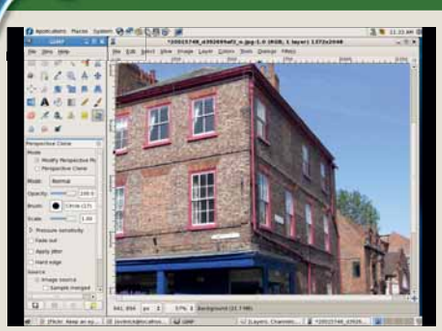
Долгое время самую достойную поддержку управления цветом среди открытых инструментов обеспечивал проект *Little CMS*, ныне интегрированный в *GIMP*. В настоящее время отсутствие поддержки CMYK в *GIMP* довольно сильно ограничивает его полезность для вывода на печать, но перемены грядут...

› Марти Мария [Marti Maria] доблестно поддерживает эту свободную библиотеку.

Инструмент Perspective Clone

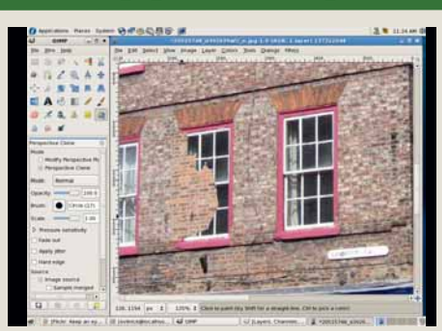
Эта новинка может показаться слегка специфической, но бывают ситуации, особенно при доводке изображений, когда она сумеет доказать свою полезность. Инструмент в точности подобен Clone, но искажает перспективу в зависимости от положения клона относительно заданной перспективы. Слишком запутанно? Да нет, но какое-то время поразбираться придется. Этот инструмент – плод Google Summer Of Code 2006, задуманный и реализованный студентом Педро Алонсо [Pedro Alonso]. Как первый набросок он отнюдь не плох, но уже планируются поправки и улучшения, включая возможность предпросмотра. Правда, остается вопрос, заслуживает ли этот инструмент иконки на вашей панели инструментов: она и так малость захламлена (и там уже есть три разных иконки для трех видов операций клонирования). »

Шаг за шагом: Клонируем с перспективой



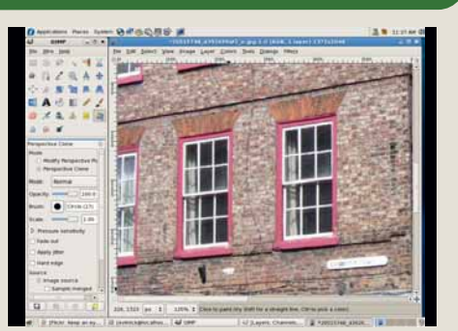
» Перспектива

Сперва настройте перспективу. Перетащите углы прямоугольника Perspective, чтобы сформировать проекцию перспективы. (См. здесь края стены).



» Выберите источник

При нажатом Control щелкните на рисунке, чтобы отметить начальную точку для клонируемого материала. Для хорошо определенного объекта лучше выбрать что-то типа угла.



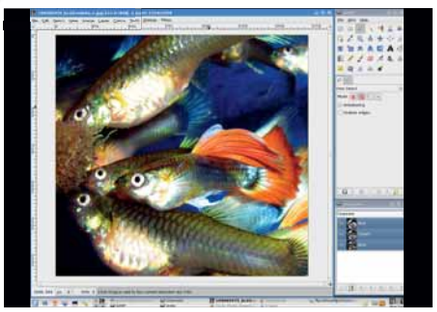
» Клонируйте!

Теперь просто зарисовывайте, пока клонирование на будет завершено. Как вы можете видеть здесь, окно клонировано и сохранило перспективу стены.

» Инструмент Foreground Select

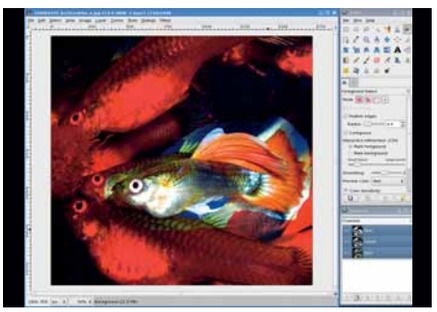
Этот новый инструмент явно один из самых обсуждаемых в версии 2.4. Основанный на технологии Simple Interactive Object Extraction (*SIOX*, www.siox.org), Foreground Select предоставляет простой способ выполнить «вырезку» (cut-out) в *GIMP*. В нем уже есть масса инструментов для отделения объекта от фона, но этот ну очень крут! И в большинстве случаев он значительно сократит время и труды на получение вырезки, особенно украсив жизнь тем, чей опыт обработки графики пока небогат: использовать его не сложнее, чем вчерне обвести объект большой толстой кистью.

Шаг за шагом: Вырезаем с помощью нового Foreground



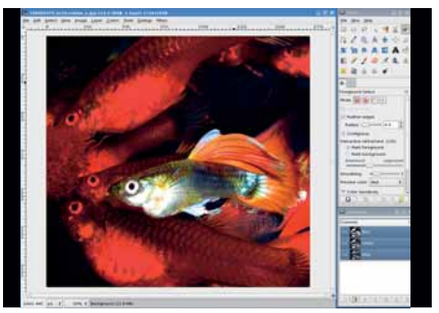
1 Выберем свой рисунок

Это сложное изображение для инструмента Foreground Select. Нам нужна ближайшая к нам рыбка, но здесь их целая стая, и все с похожей окраской. Форма у рыбки неправильная, да еще она имеет веретенообразные плавники.



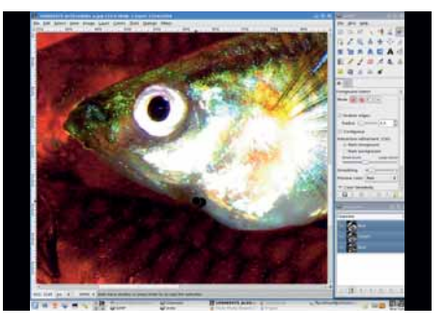
2 Заарканим объект

После выбора Foreground Select вас попросят обвести объект с помощью инструмента Lasso. Слишком точным быть не обязательно, но объект надо охватить инструментом полностью, не то придется начинать сначала.



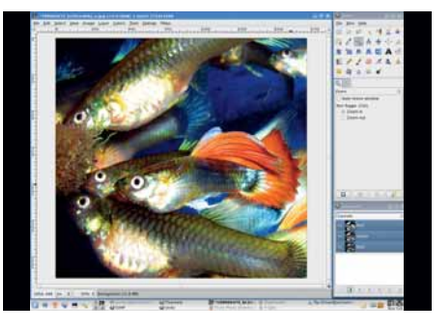
3 Просмотрим результат

По завершении работы с Lasso программа выдаст наилучшее предположение об объекте. Можно видеть, что работа не так уж плоха – рыбка вчерне выделена, хотя кое-что осталось снаружи.



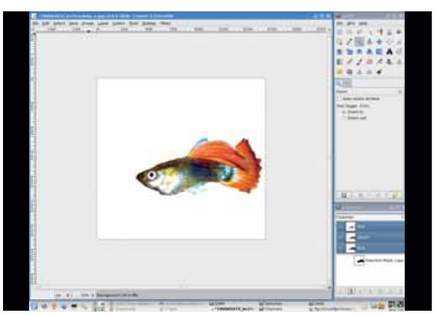
4 Улучшим его

Теперь можно повысить качество выборки, закрасив толстой кистью области, которые мы определенно считаем частью изображения (или фона – это если вы щелкнете по соответствующей радиокнопке в свойствах инструмента).



5 Посмотрим

Каждый раз, когда вы что-то рисуете, программа пересчитывает наилучшее предположение о желаемом объекте. Можете улучшать его сколько душе угодно, но не увлекайтесь – некоторые кропотливые операции, если нужно, можно будет выполнить по старинке.

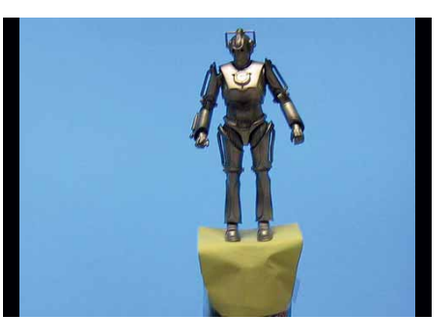


6 Вырежем

Когда выделение вас устроит, нажмите Enter. Инструмент закроется, и вы останетесь с выделенным объектом. Чтобы получить объект сам по себе, просто обратите выделение и очистите остальную часть изображения.

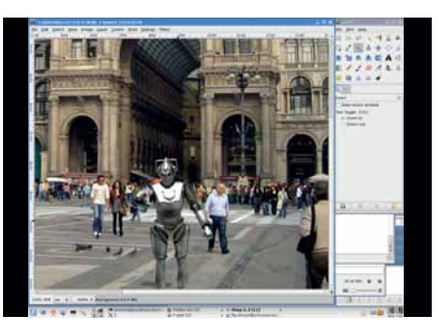
7 Поиграем!

Теперь объект ваш: делайте с ним все, что хотите. Можете продолжить улучшать выделенный объект, или использовать Foreground Select для усиления некоторой детали, или применить фильтр только к чему-то одному.



8 Попробуем простой рисунок

Наш тестовый рисунок использовался для демонстрации качества инструмента Foreground Select. Если вы хотите упростить себе жизнь, вам следует фотографировать объекты для вырезания так, чтобы их было проще вырезать, как показано здесь.



9 Вставим и растушуем

Легкая растушевка краев вырезки может сильно облегчить ее правдоподобное комбинирование с фоном, но остерегайтесь потери четкости краев там, где она должна быть – не размазывайте слишком сильно.

Мечты сбываются

Долгожданные функции добавятся в GIMP 2.6.

При всех отличных новых функциях релиза 2.4, остается чувство, что во многих отношениях разработка GIMP топчется на месте, и ряд фундаментальных проблем, как вы увидите, еще предстоит решить.

Небольшой опрос пользователей *GIMP*, проведенный *Linux Format* в сети, выявил устрашающий список желаемых функций. Без некоторых из них *GIMP* не может считаться серьезным выбором для определенного типа графических работ. Вкратце изложим наиболее популярные требования.

Поддержка CMYK

Из цветовых моделей *GIMP* в настоящее время поддерживает только градации серого, индексированные или RGB-цвета. Этого достаточно для большинства творческих работ – иконок, web-графики и даже операций с фотографиями, поскольку мир фото стал цифровым. Однако есть большой и очень важный сектор графики, не охваченный этими цветовыми моделями: печать.

Экранная графика аддитивна – различные цвета спектра смешиваются для получения желаемого оттенка. В случае печати цвета рассматриваются в отраженном свете и процесс становится субтрактивным – чернила поглощают часть спектра и отражают остальное, что и видит смотрящий. Здесь, пожалуй, не стоит вдаваться в подробности, но вам следует знать, что гамма (жаргонный термин, используемый любителями искусства для обозначения всего диапазона доступных цветов) цветовой модели для печати очень отличается от таковой для экрана – в пространстве RGB есть множество цветов, которые просто невозможно напечатать.

Другая проблема заключается в том, что при печати реально используются чернила голубого, пурпурного, желтого и черного цветов (он же «ключевой», 'key'). Отсюда и CMYK – Cyan, Magenta, Yellow, Key. Чтобы рассчитать правильную дозировку чернил, рисунок в RGB (Red, Green, Blue – красный, зеленый, синий) необходимо конвертировать,. А сможет ли ваш принтер выполнить это преобразование правильно? Или драйвер вашего принтера? Вообще-то для преобразования применяется стандартная формула, проецирующая RGB-изображения на доступный для печати диапазон CMYK. Но для конкретного изображения это может оказаться не лучшим вариантом, и часто приводит к затемнению цветов и потере контраста на некоторых участках. Всегда намного лучше доверить преобразование самому дизайнеру. А если ваше изображение уже в CMYK, вряд ли вы захотите с потерями конвертировать его в RGB только для того, чтобы на более поздней стадии преобразовывать обратно.

Хорошая новость: *GIMP 2.6* строится на движке *GEGL*, без проблем справляющемся с обработкой изображений в CMYK. Пока не все ясно с поддержкой различных цветовых пространств (см. «Что за штука... *GEGL*» в LXF83 про цветовые модели и пространства). Но все же лед тронулся. Ждем!

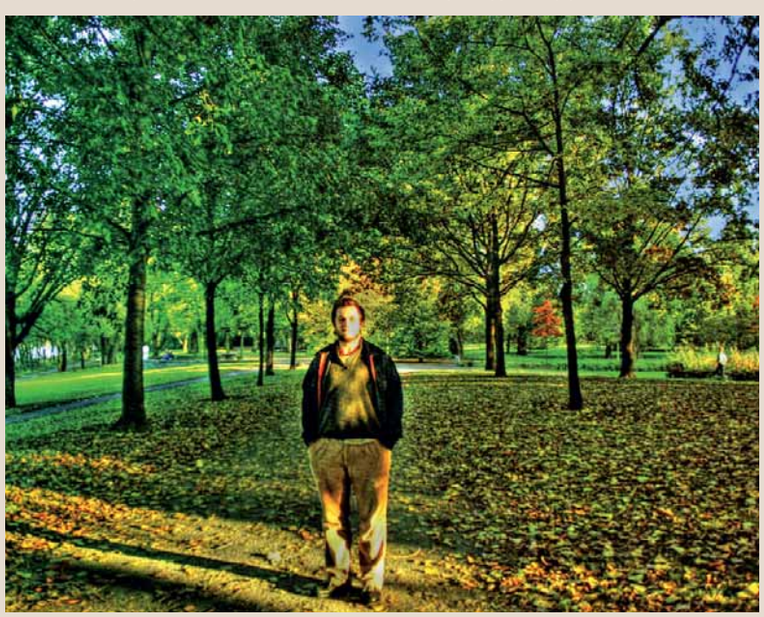
Поддержка формата RAW и разрядность цвета

Распространение цифровых фотокамер породило новые файловые форматы – для файлов, используемых самой камерой при создании знакомых нам JPEG-изображений. Эти форматы слегка различаются, но в общем и целом однотипны. Каждый формат файла просто включает все «сырые» данные напряжений, снимаемых с матрицы камеры. Обычно это 12 или 14 бит на пиксель, на цвет, т.е. заметно больше, чем 8 бит на цвет, используемые в JPEG. Внутреннее ПО камеры преобразует эти данные в JPEG, подгоняя по ходу все, что нужно – например, применяя логарифмическую кривую для подгонки контраста, повышая резкость участков за счет увеличения контраста по краям и используя прочие хитрости, чтобы картинка выглядела круто.

Искусство GIMP: Гуннар Гримнес

К сожалению, нынешние ограничения на разрядность цвета в *GIMP* препятствуют экспериментам с «сырыми» данными изображений с цифровых камер, и это огорчает желающих испытать эффекты HDR (High Dynamic Range) или оптимизировать преобразования «сырых» данных в финальные изображения. Гуннар Гримнес [Gunnar Grimnes] использует *GIMP* и целый ряд других открытых инструментов для экспериментов с изображениями. Показанное здесь HDR-изображение было на самом деле создано в *CinePaint*, отпрыске *GIMP*, ориентированном на киноиндустрию: он может загружать «сырые» данные с большей разрядностью цвета.

Проще говоря (а если вам нужны детали, можете заглянуть на http://en.wikipedia.org/wiki/High_dynamic_range_imaging), HDR – это способ захвата большей информации о тоне: деталей по всей шкале градаций изображения. И его можно использовать для получения некоторых крутых эффектов.



Понятно, что многим хочется получить больший контроль над выполнением этого JPEG-преобразования. Камеры, поддерживающие сохранение в режиме RAW, обычно создают TIFF-файл с частично преобразованными данными, но его загрузка требует от ПО поддержки большей разрядности цвета. Разрядность цвета весьма существенна для всего, что делает *GIMP*, так что простой доработкой поддержку RAW и других форматов с большой разрядностью цвета (например, для эффектов динамического диапазона – High Dynamic Range, HDR) не получить. *GEGL* решит и эту проблему, но если ждать невмочь, взгляните на ответвление *CinePaint* (*см. врезку «Раскольники!» далее*), которое умеет работать с HDR и другими интересными вещами уже сейчас...

Коррекционные слои, эффекты слоев

Неразрушающее редактирование позволяет вам, скажем, применить фильтр размывания к изображению, кадрировать его, скорректировать цвета, а затем вернуться назад и отменить выполненное размывание. При стандартном линейном редактировании, которое мы склонны использовать в графике, такое невозможно, но есть способы сделать сказку былью. *Adobe Photoshop* реализует некоторые из этих функций с помощью коррекционных слоев, но *GEGL* может обеспечить гораздо большее.

GEGL: избранный

Мы уже не раз упоминали *GEGL* в нашем обзоре списка желаемых функций *GIMP*. Для новичков поясним, что *GEGL* – это совершенно »

» новый базовый движок для выполнения операций над пикселями, на котором строится *GIMP 2.6*, и к нему обращено довольно много надежд художников и дизайнеров. Создавался он, чтобы обеспечивать функциональность *GIMP*, но к его API можно легко получить доступ из другого кода – фактически, уже разрабатываются привязки для Python и Ruby.

Сам по себе *GEGL* никакой обработки пикселей не выполняет: вся мощь обеспечивается подключаемыми модулями, а операциями на уровне пикселей занимается библиотека *babl*. *GEGL* управляет этими данными структурированным способом, используя так называемый нециклический направленный граф (Directed Acyclical Graph). Изображений как таковых в *GEGL* не существует: по большей части он работает с потенциалом изображения, выстраивая структурированную коллекцию узлов, каждый из которых отвечает некоторой операции обработки. Таким образом, код содержит рецепт достижения результата, а не сам результат. Можно подключить выходной узел для показа текущего состояния изображения на экране, при этом большая часть обработки будет отложена, пока не понадобится выполнить окончательный рендеринг.

Реализуется потенциал оптимизирования операций (выполняются только те из них, что непосредственно сказываются на выводе), а заодно мы получаем больший контроль над процессами, составляющими генерацию изображения. Это открывает путь таким возможностям, как коррекционные слои и другие атрибуты неразрушающего редактирования.

У *GEGL* интересная история. Он возник как часть ответвления *FilmGIMP*, позже развившегося в *CinePaint*. Работы над ним были начаты в 2000 г., но импульс не получил поддержки, и проект вскоре заглох. В 2003-м его попытались возродить, но в полной мере он вновь активизировался только в 2005 г., когда первоначальная кодовая база была почти полностью переписана. В результате огромной работы, проделанной Ойвиндом Коласом [Øyvind Kolas] и другими разработчиками проекта *GIMP*, ликвидированы прорехи в коде, и проект наконец задышал полной грудью.

GEGL еще не достиг стадии финального релиза, хотя текущая версия на момент подготовки статьи, 0.0.10, уже лежит в основе нескольких испытательных приложений. Насколько он далек от выхода полностью стабильной версии? Трудно сказать, но поскольку следующий релиз *GIMP* будет основан на *GEGL*, а ожидается он не позднее 2008 г., то, вероятно, осталось не так уж много. Ойвинд Колас сказал нам: «На сегодняшний день задача сводится к тому, чтобы зафиксировать API и заставить его работать. Эта работа почти завершена. Оптимизация кодовой базы не выполнялась, и, вероятно, также будет сделана немного погодя. Следующий шаг после публичного API – достижение хорошей внутренней архитектуры.»

За прошедший год Колас вложил в *GEGL* немало труда, но поскольку он собирается скоро приступить к кандидатской диссертации, его вклад в проект в будущем может снизиться (если никто не захочет проспонсировать его работу над диссертацией по теме *GEGL*!). Впрочем, формат кода сейчас понятен, и есть надежда, что появятся и новые люди, желающие помочь разработке.

Что же касается проекта *GIMP* в целом, то раз используется движок *GEGL*, переработанный и исправленный, вскоре мы можем увидеть взрыв новых функций и гораздо более частые обновления – если разработчики изыщут время для этого.

Как и в случае со многими открытыми проектами, мы, в общем, получаем то ПО, которого заслуживаем. *GIMP* – жизненно важная часть Linux, претендующего на звание полноправной настольной ОС, и регулярно используется миллионами. Почему бы не выкроить время, не посмотреть, чем вы можете содействовать сообществу *GIMP*, и не сделать последующие десять лет *GIMP* эффективнее, чем прошлое десятилетие? LXF

Раскольники!

Ответвление *FilmGIMP* произошло в 2002 г., когда различные киностудии проявили интерес к использованию ПО, основанного на Linux. Потребность в поддержке высокоточных форматов файлов (OpenEXR, Cineon и т.д.) означала, что любой используемый графический инструмент обязан поддерживать большую разрядность цвета и различные цветовые пространства. Проект *FilmGIMP* взял существующий код *GIMP* и переписал движок обработки графики.

Этот проект имел большой успех в Голливуде и применялся во многих известных фильмах (в том числе «Планета обезьян» и «Человек-паук»). Но как графический редактор, *CinePaint* [в который развился *FilmGIMP*, – прим. пер.] испытывает недостаток во многих инструментах, к которым мы привыкли. Не воспользовался он и достижениями пользовательского интерфейса. Так что хотя *CinePaint* может быть полезен для конкретных задач, он не обладает функциональностью *GIMP*.

CinePaint все еще развивается (**www.cinepaint.org**), и сейчас продолжается работа над новой версией.

Интервью: Свен Нойман

Лидер проекта GIMP очертил нам дальнейшие планы.

Linux Format: Далеко ли, по вашему мнению, до 2.6? Между 2.2 и 2.4 был довольно большой зазор.

Свен Ньюман: Мы планируем выпустить 2.6 всего через несколько месяцев после 2.4. Набор планируемых изменений ограничен. В основном это внутренние вещи. Пользователи заметят разницу, но пока я не хочу обещать какие-нибудь выдающиеся функции в 2.6.

LXF: Что, по вашему мнению, будет ключевой функцией в 2.6?

СН: Внутри мы начнем использовать *GEGL*, и мы портируем наш код отрисовки на *Cairo*. Первый наконец-то позволит нам получить большую разрядность цвета и цветовые пространства, отличные от RGB. Но еще нужно посмотреть, что из этого проявит себя в 2.6.

Рендеринг на базе *Cairo* сделает доступными инструменты для более тонкой прорисовки в окне изображения. Например, представьте себе инструмент Path, рисующий сглаженные кривые Безье. Эта инфраструктура позволит значительно улучшить удобство наших инструментов.

LXF: В чем *GIMP* нуждается больше всего? Разработчики, технические писатели, тестеры...?

СН: Разработчики! Работы полно, а занимается ею небольшая группа активных разработчиков, и только в личное время. Помните проект *GIMP*, спонсируемый OpenUsability [финансируемая Open Society Institute инициатива по поддержке студентов, занимающихся эргономикой, для работы над свободными и открытыми программными проектами]?

Он жив-здоров. Уже почти полгода Питер [Сиккинг, Peter Sikking] и Камила [Гедройц, Kamila Giedrojc] вкалывают над оценкой *GIMP* и пишут спецификации для пользовательского интерфейса следующего поколения. Скоро мы увидим результаты их работы, и я уже могу сказать, что нам потребуется немалая помощь в реализации нашего видения *GIMP 3.0*.

LXF: А чем могли бы помочь пользователи этой программы?

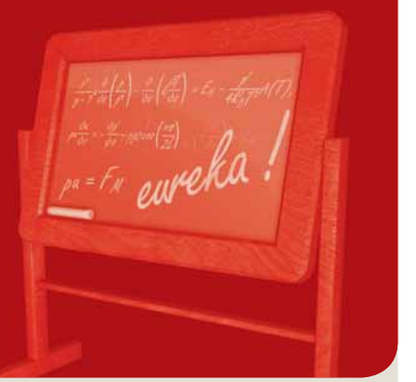
СН: Пусть подключаются. Нам нужны не только тестеры, нам нужно больше людей, работающих над пользовательской документацией, да и сайту нашему нужен капитальный ремонт!

Подключайтесь

Мы надеемся, что эта статья показала, что готовит для вас *GIMP*. У графики в Linux, определенно, перспективное будущее. Между тем, почему бы не заглянуть на **www.gimp.org** и не посмотреть, чем вы можете помочь приближению этого будущего? Команда разработчиков куда доброжелательней, чем вы, вероятно, думаете, и есть множество способов помочь ей. А те, кто просто хочет поэкспериментировать, могут получить последнюю версию для разработчиков с **www.gimp.org/source**.

LINUX FORMAT

Учебники



Наши эксперты помогут вам с любым приложением Linux

ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Пора легализоваться

«Ржав вновь задумался. У него был вид газонокосилки, столкнувшейся с профсоюзом травинок»

Терри Пратчетт. Патриот

Маски-шоу с целью поиска нелицензионных программ переходят из разряда страшилок в обыденную реальность. И что бы там не говорили создатели Windows, именно они являются причиной этого. Как следствие, наша система в основном работает именно на них.

С другой стороны, а чего нам бояться? У нас всё чисто — GNU/Linux и никакого контрафакта. Но пофантазируем, ведь у страха глаза велики: а что, собственно говоря, является доказательством его легельности? Ну да, в конце концов сервер вернут и, возможно, даже извинятся – месяца через два. Вам от этого легче?

Одни вопросы. Какие внешние признаки указывают на лицензионность ПО? Что мы знаем про это? Что знают те, кто приходят проверять? Может быть, есть какие-нибудь признаки, которые успокоят их? Может быть, удастся выработать подходящее решение внутри сообщества и взять его всем на вооружение? Тем более, что нечто похожее на подобное решение уже начинает кристаллизоваться. На сайте «Российской неорганизации поддержки свободного ПО» выложены сканы нотариально заверенного перевода GPL v.2 (http://www.rfsi.ru/node/79). Это сделано только для примера, что такое возможно. Зачем это нужно – отдельная история. Но видно, что «профсоюз травинок» потихоньку начинает свою организационную деятельность.

Я вынужден поделить свою колонку на две части, потому что 19 марта 2007 года в возрасте 82 лет умер Джон Бэкус, которого по праву называют создателем Fortran. Fortran (Фортран) является первым рабочим высокоуровневым языком программирования, модификации которого активно используют до сих пор. Этот человек определённо что-то изменил в нашем мире.

E.M.Baldin@inp.nsk.su

В этом выпуске...

56 Азбука безопасности
Linux гораздо защищеннее Windows, но чтобы быть действительно в безопасности, необходимо хорошо представлять себе, что происходит. **Энди Ченнел** предлагает краткий обзор защитных механизмов Unix.

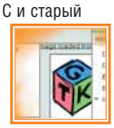
64 Подстройка SugarCRM
Клиенты Риты отличаются от клиентов Сью, чьи клиенты отличаются от клиентов Боба. К счастью, у нас есть **Марк Бейн**, который может все это разрулить...

60 Улучшаем RSS с помощью XSLT
XML делает свое дело, но редко выглядит привлекательно. **Рейчел Проберт** покажет, как использовать XSLT для создания красивой RSS-ленты.

68 GUI в Mono
Готовы попробовать новое? **Пол Хадсон** проведет вас по всем этапам создания графического RSS-агрегатора в Stetic!

72 Дебианизация всей страны
Почему бы не сделать что-то полезное для сообщества, собрав столь нужный людям deb-пакет? Особенно теперь, когда у вас под рукой есть учебник **Грэма Моррисона**...

76 Новый Glade
Stetic и C# – это хорошо, а C и старый добрый Glade – лучше! **Андрей Боровский** покажет, как создавать интерфейсы GTK+ в стиле новой школы.



80 Программирование с ncurses
Окна бывают не только у графических приложений! **Андрей Боровский** учит ваять красивые и удобные интерфейсы для консольных программ.

84 Сессии и фильтры Java EE
Редкое web-приложение обходится без аутентификации. О том, как реализовать ее средствами Java EE – в статье **Антона Черноусова**!

88 Допглавы PostgreSQL
Резервирование, репликация, автовакуум – все, что вы хотели знать, но боялись спросить, в заключительной статье серии **Евгения Балдина**.

94 Презентация в LaTeX
Если вам проще писать, чем рисовать, то OOo Impress'у вы предпочтете LaTeX. Кстати, яркие презентации можно сохранять не только в формате PowerPoint, и **Евгений Балдин** знает, как.

98 Шагающий пингвин
Андрей Прахов завершает серию уроков Blender рассмотрением скелетной анимации.

Совет месяца: Linux sux

Команду *su* чаще всего вызывают из консоли, чтобы отредактировать какой-нибудь общесистемный конфигурационный файл, но иногда возникает необходимость повысить привилегии для запуска графического приложения. Попробовав сделать это, вы получите такую ошибку:

```
Xlib: connection to ":0.0" refused by server.
```

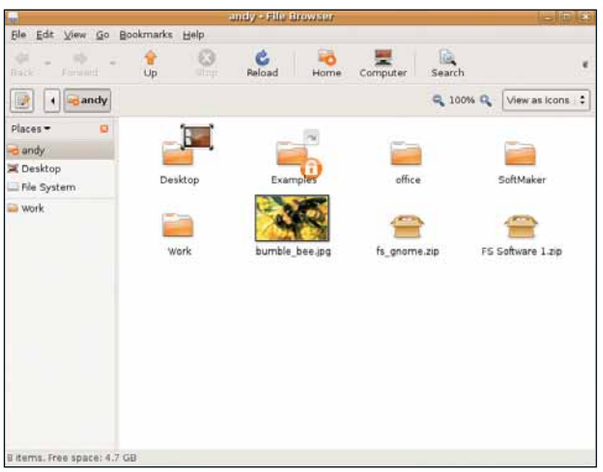
Проблема заключается в том, что при переключении пользователя ключи, используемые для аутентификации вашей *X*-сессии, становятся недействительными. Сегодня эта мера безопасности может показаться излишней, но она уходит корнями в те времена, когда *X*-сессии были преимущественно сетевыми. Проблема имеет 2 решения.

Первый подход: обеспечить аутентификацию на X-сервере для новой (привилегированной) учетной записи. Для этого используется команда *xhost*. Выполните **xhost +local:local** от имени вашего обычного пользователя и вы увидите надпись: «non-network local connections being added to access control list». Теперь новый пользователь сможет запускать любое графическое приложение на том же *X*-экране. Если вы хотите включать эту возможность каждый раз, когда вы открываете *X*-сессию, следует добавить соответствующую команду в файл **.bashrc** – тогда она будет выполняться при старте новой копии оболочки *Bash*. Неприятным побочным эффектом данного решения является возникновение аутентификационной петли при попытке переключиться с помощью *su* обратно на вашу учетную запись. Используйте команду *exit*.

Второе решение еще проще. Используйте команду, альтернативную *su*, которая автоматически передает реквизиты вашей X-сессии. Она называется *sux*, и ее придется доустановить из репозиториев вашего дистрибутива вручную. После этого, для переключения учетной записи будет достаточно набрать **sux newuser**. Вы сможете запускать графические приложения без какой-либо дополнительной настройки.

Система: Рули

Почему Linux столь защищен? **Энди Ченнел** считает, что благодаря правам доступа, привилегиям и силе. И он прав.



› Безопасность начинается с /home, где вам следует определить права доступа для других пользователей.

Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется еще со времен Dragon 32.

Сила вертит этим миром. Правда, экономисты уверяют, что деньги, а романтики предпочитают думать, что любовь, но это в чистом виде сила. В XXI веке, когда так много важной персональной информации хранится на жестких дисках и в компьютерах, эта сила иногда может быть названа одним словом: владение [owned]. Намеренная опечатка в этом слове – 'pwned' – означает, что кракер, а не хакер (*см. внизу справа врезку «Кракеры против Хакеров»*), получил контроль над вашим ПК и может использовать его в своих целях.

Гнусными целями могут быть: рассылка спама по миллионам адресов, отправка вируса всем, кто записан в вашей адресной книге, хранение и передача аморальных или противозаконных изображений или использование вашего браузера для накрутки счетчиков посещений некоторого web-сайта. Это не просто мелкие жулики, интересующиеся вашим ПК: кракинг – большой бизнес. В мае 2006 г. Sony BMG Music Entertainment (да, та самая Sony!) была уличена в распространении руткита через DRM-приложение, записанное на одном из ее музыкальных CD и выпущенное 12-миллионным тиражом. Любой осведомленный об этом кракер мог получить удаленный доступ к ПК. Опасности сетевого вторжения были подвергнуты миллионы пользователей.

Это, конечно, одна из причин, почему мы используем Linux; в мире Linux тоже встречаются руткиты, но они менее распространены, более вероятно, что они будут обнаружены и деактивированы, а благодаря организации ОС Linux им труднее нанести серьезный ущерб [это распространенное заблуждение. Руткит, действующий как часть ядра, способен на все, как в Windows, так и в Linux, – *прим. ред.*]. Однако функции, обеспечивающие защищенность Linux, заботятся в основном о самой ОС, тогда как среднего пользователя скорее волнуют личные данные, среди которых может быть финансовая информация и, чем дальше, тем больше, обширные хранилища фотографий, видео и документов. Важны оба этих аспекта: тот, кто прорвется в ваш компьютер, может установить теневую программу слежения за клавиатурой (keylogger), которая в фоновом режиме будет записывать все, что вы набираете (имена пользователей, пароли, номера счетов и т.д.), то есть уязвимость ОС может привести к потере или несанкционированному использованию ваших данных.

Исправление проблем безопасности в Linux – работа для профессионалов, а мы, скромные конечные пользователи, должны довольствоваться загрузкой и установкой соответствующих заплаток по мере их появления. Но это не значит, что мы бессильны. На данном уроке мы рассмотрим несколько методов защиты, а заодно разберемся, как и почему пользователи Linux поделены на рядовых и привилегированных. Мы также обсудим способы блокирования файлов и папок, запрещающие другим пользователям в многопользовательском окружении – в офисе, школе или дома – изменять их или вообще знать об их существовании.

Корень проблемы

В Linux вы можете встретиться (и работать от имени) пользователя одного из двух типов: рядового (user) и привилегированного (root). Давайте рассмотрим, что может и что не может делать каждый из них. С точки зрения безопасности, root важнее, вот почему опытных пользователей раздражает, когда издатели дистрибутива решают, из соображений удобства, создавать обычных пользователей с правами root. Такое было в Linspire, но теперь при установке вы можете выбрать, создавать ли пользователя root: пользуйтесь этим.

Месяц назад Мы изучали способы установки и обновления RPM-пакетов.

пользователями!

Вообще-то, root – это администратор компьютера, и вам необходим пароль root для выполнения задач типа установки программ, изменения прав доступа к каталогам за пределами вашего пространства **/home/имяпользователя** и изменения параметров устройств. Этот пароль назначается вами при установке дистрибутива, и он должен быть стойким. Если ваш пароль слабоват, большинство дистрибутивов предупреждает об этом.

Кстати, Ubuntu использует компромиссную систему, схожую с Apple OS X, где пользователям временно предоставляются привилегии root, когда они вводят свой собственный пароль. Эта система называется *sudo*, и она неплохо работает: ведь кракеру, пожелавшему взломать вашу машину, придется узнавать и имя пользователя, и пароль, тогда как в противном случае будет достаточно подобрать пароль root.

Если вам захочется добавить традиционную учетную запись root в Ubuntu (что разработчики дистрибутива не рекомендуют делать), просто откройте консоль и наберите **sudo passwd root**. У вас спросят ваш собственный пароль, затем новый пароль и его подтверждение. Сделав это, можете открыть консоль, набрать **su** и ваш пароль для получения полного доступа к системе. Однако *sudo* остается активной, и когда при установке программ запрашивается пароль, имеется в виду пароль пользователя, а не root. Вы можете сделать так, чтобы *sudo* запрашивала пароль root, отредактировав файл **/etc/sudoers**. Этот файл можно открыть в *Gedit* или *Kate*, запустив эти приложения с привилегиями root (см. ниже). Найдите в тексте файла строку, начинающуюся с 'Defaults', и добавьте ключевое слово 'rootpw'.

Определив пользователя root, можете завершить сеанс и набрать имя **root**, а затем тот самый пароль в окне входа в систему. Вход в режиме root можно запретить, набрав **su passwd -l root**, но если вы сделаете это при настройке *sudo* на использование пароля root, то не сможете нормально администрировать свою систему. Пожалуйста, помните, что все-таки разделение на пользователей и суперпользователей имеет свои причины: будьте осторожны.

Если вы прикипели к *sudo*, но хотите обеспечить другим пользователям удобство установки программ и совместного использования файлов, наделите их соответствующими правами, войдя в System > Administration > Users And Groups [Система > Администрирование > Пользователи и группы], выбрав соответствующего пользователя и нажав кнопку Properties [Свойства]. Выберите вкладку User Privileges [Привилегии пользователя] и установите опцию Executing System Administration Tasks [Выполнение задач системного администрирования]. Теперь, запустив *sudo* во время работы, другой пользователь сможет ввести свой собственный пароль.

Кракеры против Хакеров

За последнее десятилетие «хакер» стало синонимом компьютерных преступников, но это слово с большой предысторией, имеющее больше общего с наладкой, чем со злодеянием. Большинство разработчиков программ опишут свою деятельность как хакинг, как и те, кто пытается воткнуть Linux во все устройства от Nintendo Wii до робота-пылесоса Roomba. Другая сторона медали – это кракеры, те, кого больше интересует поиск уязвимостей систем и злоупотребление ими для своего удовольствия (денежного или личного).

› Подлинно многопользовательская система означает, что у каждого члена вашей семьи – своя собственная часть компьютера.

Идеальный пароль

На элементарном уровне безопасность вашего компьютера определяется ее слабейшим звеном, и часто это пароль. Выбирая его сложным для взлома, вы защищаете себя от локальных (связанных с физическим доступом к машине) и удаленных (доступ по сети) атак. Хороший, безопасный пароль должно быть трудно взломать, так что избегайте дней рождения, имен детей, и тем более явных слов вроде password. Кроме того, большинство людей рекомендуют использовать комбинацию как минимум шести символов, и строчных и заглавных, а также числа: это не даст кракеру осуществить лобовую «атаку по словарю».

Один из способов – взять начальные буквы запоминаемой фразы (например, Пока Коровы не Вернутся Домой) или песни и вашу дату рождения, так что в итоге получится ПКнВД130271 [имейте в виду, что использовать кириллицу в пароле не рекомендуется, – *прим.ред.*]. Можно также использовать имя и две даты, например, Mozart17561791 – заодно освежите в памяти историю! Если вам неохота удерживать в голове строки практически случайных символов, почему бы не объединить два слова, по типу PaperGorilla – и запомнить легко, и взломать труднее, чем одно слово [правильнее сказать, «незначительно труднее» – поэтому таких паролей также следует избегать, – *прим. ред.*].

Итак, у нас есть пользователь root и хороший сильный пароль для защиты от вторжения со стороны других пользователей. Вход root через приглашение KDE или Gnome не отличается от входа обычного пользователя, но он все же не рекомендуется к использованию. Гораздо лучше работать под своей учетной записью, а когда понадобится, давать себе привилегии root на время выполнения конкретных задач. Обычно это автоматизировано: при установке программ, например, в какой-то момент у вас запрашивается пароль root. Однако иногда может потребоваться сделать это вручную, например, в ситуации, когда вы пытаетесь собрать и установить программу из исходных текстов. Тут нужно открыть консоль и набрать **su**, нажать Enter, затем »

Скорая помощь

Перемещайтесь в командной строке при помощи команды *cd*. Например, находясь в **/home**, наберите **cd documents** – и переместитесь в каталог **/home/document**s. Команда *ls* выведет список содержимого каталога.

» ввести пароль root. После этого все, что вы набираете в консоли, будет выполняться от имени суперпользователя (root).

Поскольку приложения также запускаются из командной строки (путем набора имени приложения), можно открыть текстовый редактор вроде *Kate*, способный редактировать системные файлы, или запустить файловый менеджер вроде *Konqueror* или *Nautilus*, умеющий копировать, вставлять или редактировать любые файлы в системе. Однако это не очень здорово, поскольку может привести к осложнениям, если позже в систему попробуют зайти другие пользователи. Лучший способ выполнения таких задач – использовать *gksudo* (для пользователей Gnome) или *kdesu* (для KDE). При этом запускается отдельный экземпляр приложения с указанными правами, что уменьшает вероятность случайного вторжения в другие области операционной системы.

Чтобы сделать это, нажмите Alt+F2 для открытия диалога Выполнить команду и введите

```
gksudo appname
```

или

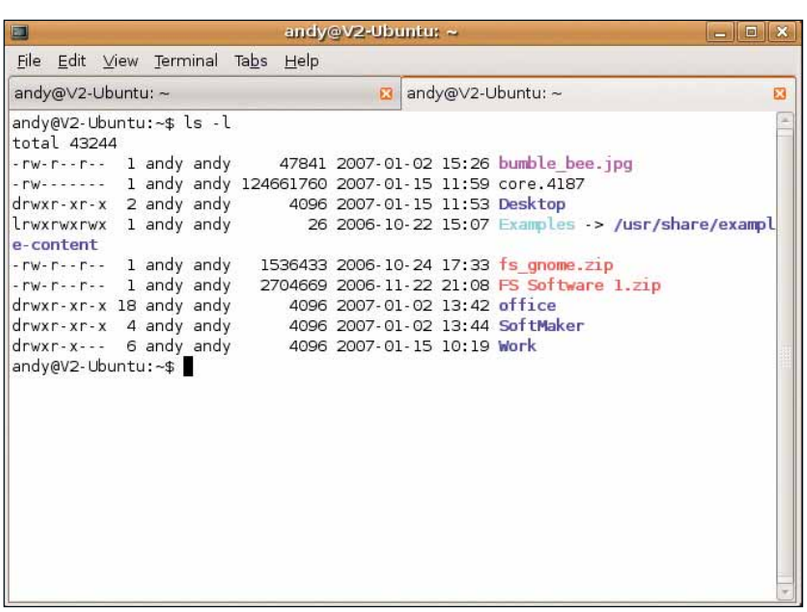
```
kdesu appname
```

где **appname** – имя запускаемого приложения. Теперь это приложение имеет права root, то есть пользователь может открывать любые файлы, включая системные или файлы настроек, и изменять их, но эти привилегии будут отменены при закрытии приложения, тогда как при использовании командной строки привилегии root заканчивают свое действие только при закрытии терминала.

Другой тип пользователя – это, хм... пользователь (user). Для компьютера, пользователь – отдельная сущность, с собственными настройками, рабочим столом и каталогом внутри **/home** для размещения личных документов, фотографий и прочего. В своем домашнем каталоге (где имеется подкаталог с именем **/Desktop**, хранящий файлы, помещенные на рабочий стол), пользователь может вытворять все, что угодно, но он не может изменять какие-либо другие файлы – включая домашние каталоги других пользователей. Это достигается посредством прав доступа, определяющих, кто и что может делать с каждым файлом и каталогом на компьютере.

Как вы понимаете, права доступа управляются не единственным образом, и это важно для совместного пользования информацией на компьютере или в сети, особенно если у вас дома больше одного пользователя. Например, вы можете запретить детям доступ к вашему рабочему каталогу, но позволить им пользоваться коллекцией музыки; или совместно использовать библиотеку фотографий, куда каждый пользователь сможет выкладывать свои цифровые снимки для всей семьи.

Права доступа обычно устанавливаются в три различных состояния, но есть еще и дополнительные опции.



› В командной строке легко отобразить права доступа к папке.

›› None (Нет доступа) Другие пользователи не могут получить доступ к содержимому каталога. Это наиболее безопасное и/или параноидальное состояние.

›› Access files (Доступ к файлам) Другие пользователи могут заходить в каталог и открывать файлы, но не изменять или удалять их. Это состояние принято было называть 'только для чтения', но и Gnome и KDE пытаются очеловечить этот процесс, давая состояниям прав хорошие удобочитаемые имена.

›› Create and delete files (Создание и удаление файлов) Любой пользователь может добавлять, удалять, просматривать и редактировать файлы как в своем домашнем каталоге (это называется Чтение/Запись).

Есть еще опция, делающая файл (не каталог) исполняемым, ее можно использовать, когда вы загрузили приложение и хотите запустить его от имени своего пользователя. Щелкните правой кнопкой на приложении и выберите Свойства > Права, а затем отметьте опцию 'Выполнение': она находится у нижнего края окна.

Разрешение на посадку

В Linux есть несколько способов представления прав доступа к файлу или папке, и хотя новичкам удобнее свалить работу на Gnome и KDE, следует знать более стандартные способы записи, чтобы, столкнувшись с проблемой в незнакомом окружении рабочего стола, вы все-таки смогли редактировать права как эксперт. Наиболее общая форма указания прав – текстовая: состояние прав файла или каталога для его владельца, группы и других пользователей описывается комбинацией букв r, w и x. Вы, верно, догадались, что 'r' – это чтение, 'w' – запись, а 'x' – выполнение. Вдобавок, дефис (-) может означать, в зависимости от позиции, что рассматриваемый элемент – это файл, или что вам нельзя его даже трогать.

При таком методе каждый файл и каталог в системе Linux имеет настройки прав доступа, указанные десятью цифрами или битами. Вы можете увидеть их в действии в Gnome, открыв менеджер файлов и выбрав опцию Права в меню Вид > Отображаемые колонки, а в KDE'шном *Konqueror* – настроив опции вывода в виде списка в меню Вид.

Проверить права каждого файла в каталоге можно также из командной строки. По умолчанию, открывая консоль, вы оказываетесь в своем каталоге **/home**. Чтобы увидеть права доступа ко всему, что здесь есть, наберите **ls -l**. Результат будет примерно таким:

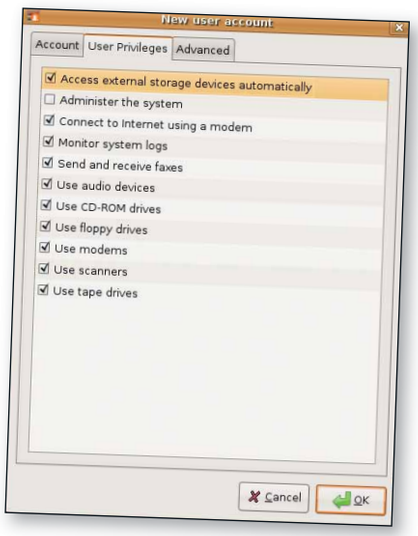
```
-rw-r--r-- 1 andy andy 47841 2007-01-02 15:26 bumble_bee.jpg
drwxr-xr-x 2 andy andy 4096 2007-01-15 11:53 Desktop
lrwxrwxrwx 1 andy andy 26 2006-10-22 15:07 Examples -> /usr/ share/ example-content
drwxr-x--- 6 andy andy 4096 2007-01-15 10:19 Work
```

То, что нам интересно, находится в первой части каждой записи. Первая цифра (позиция 0) сообщает, что мы видим: файл (-), каталог (d) или ссылку (l). В данном случае ссылка – это ссылка на файл или каталог в другой части системы. Следующие три цифры (позиции 1–3) – это права владельца, 4–6 – права группы, а 7–9 – права остальных.

К примеру, ссылка **Examples** (*см. выше*) и каталог **Shared**, который мы скоро создадим, имеют следующие права: lrwxrwxrwx. l означает, что это ссылка (далее будем использовать d), а rwx (3 раза) показывают, что владелец, группа и прочие могут читать, записывать и просматривать содержимое этого каталога. Для сравнения: у каталога **Work**, скрытого от всех, кроме активного в данный момент пользователя, указано drwxr-x---. Последние три дефиса говорят, что 'остальные' (пользователи, не являющиеся владельцем и не принадлежащие к группе 'andy') не могут ничего читать, записывать и просматривать в этом каталоге. Остальная информация, слева направо, это владелец, группа, размер файла или каталога, дата создания или изменения и имя файла/каталога. Отметим третью запись, в которой отображаются имя ссылки и место, на которое ссылаются.

И снова в home

Linux – это действительно многопользовательская система, где каж-



› «Не будет тебе прав доступа к CD-ROM, пока не сделаешь уроки!» Или нечто подобное.

дый может иметь свои собственные настройки. Первым делом следует убедиться, что у каждого члена семьи имеется свой пользовательский профиль. Для этого примера мы используем Ubuntu, но в любом другом дистрибутиве процесс управления пользователями похожий, даже если используется KDE. Начнем с запуска приложения Управление Пользователями. В Gnome оно расположено в Система > Администрирование > Пользователи и Группы; в KDE загляните в Настройка > Настроить > KUser. Нажмите кнопку Добавить для открытия окна настроек пользователя и добавьте полное имя пользователя название учетной записи. Полное имя будет использоваться в качестве имени каталога пользователя в **/home**, а название учетной записи (оно не может содержать пробелы или символы пунктуации) нужно будет вводить в окне входа в систему. Давать всем подряд привилегии администратора бессмысленно, поэтому для большинства пользователей сохраните настройки по умолчанию. Пароли можно добавить вручную (рекомендуется), а можно сгенерировать случайный, используя опцию внизу окна.

Если вам нужны более экзотические варианты, добавьте или удалите опции на вкладке Привилегии Пользователя. Доступны такие опции, как запрет доступа к CD-приводу или внешним устройствам или запрет на соединение с Интернетом через модем. Последняя опция весьма полезна, если у вас телефонный доступ и вы хотите запретить кому попало лазить в сеть и наматывать большие счета. Настроив всех пользователей, нажмите кнопку Закрыть.

Раз у нас стало много пользователей, возможно, следует запретить им доступ к личным данным в вашем домашнем каталоге. Для этого откройте домашний каталог со своего рабочего стола, щелкните правой кнопкой на каталоге, который хотите защитить, и перейдите на вкладку Права. Загляните в Другие настройки и измените опцию Доступ к каталогу на Нет Доступа. После этого другим пользователям компьютера при попытке открытия вашего домашнего каталога будет сообщено, что у них нет прав на доступ к этому ресурсу.

Теперь создадим общий каталог, доступный всем пользователям системы – так что ведите себя в нем осторожно! Откройте терминал и наберите **gksudo nautilus**. Как описано выше, это приведет к запуску *Nautilus* с привилегиями `root`. Появится запрос на ввод вашего пароля, и *Nautilus* запустится. Выберите Файловая система/home, щелкните правой кнопкой мыши в окне и выберите пункт Создать каталог. Назовите его **Shared**. Теперь щелкните правой кнопкой мыши на новом каталоге, выберите Свойства и вкладку Права. В настройках Другие, выберите Доступ к Каталогу > Создание и Удаление файлов. Теперь любой пользователь может добавлять и удалять файлы и каталоги в **Shared**, что делает его прекрасным местом для хранения образов CD, фотографий или фильмов. Помните, что пользователь `root` имеет доступ ко всем файлам и каталогам системы и может менять права доступа, вот почему так важно, чтобы ваша `root`-запись имела действительно хороший пароль! И не давайте своей супервласти вскружить вам голову. LXF

Глоссарий

» Атака по словарю Кракер настраивает программу на подстановку в поле ввода пароля слов из словаря. Если ваш пароль – обычное слово, ваша система будет взломана.

» Группа Группы в системе Linux используются для того, чтобы не определять права для каждого пользователя в отдельности. Например, вы можете разрешить группе 'Users' доступ к некоторому файлу, и тогда эти права получат все члены данной группы.

» Права Они определяют, кто и что может делать с файлами и каталогами в системе Linux. Если у вас нет прав на доступ к каталогу, вы не сможете в него войти – как, к счастью, заодно и кракеры, и вредоносные программы (malware).

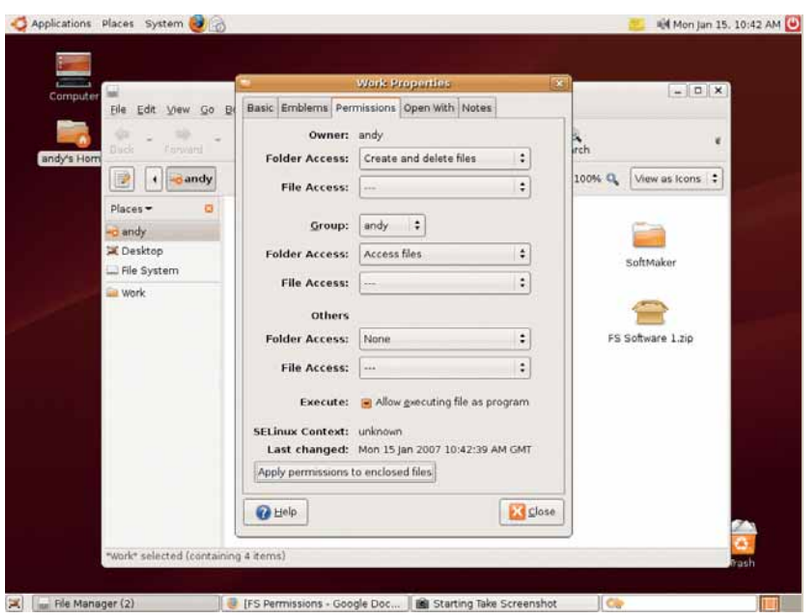
» Только для чтения Пользователь может просматривать содержимое файла или каталога, но никак не может изменить его. Правда, он может открыть документ или файл и затем сохранить его в собственном пользовательском пространстве, что даст ему права на чтение/запись.

» Чтение/Запись Пользователи могут просматривать и изменять содержимое по своему желанию.

» Root Администратор данной системы. root может производить изменения в системе – устанавливать программы, настраивать устройства и т.д. – чего не может обычный пользователь. Вы можете «стать root'ом» (получить привилегии root), открыв консоль и набрав **su**, а затем введя пароль root. Не забудьте закрыть консоль root по завершении работы или набрать exit для отмены привилегий root.

» Пользователь (User) Обычный пользователь системы Linux. Поскольку Linux – настоящая многопользовательская ОС, каждый, кто постоянно работает за данным компьютером, имеет отдельную учетную запись (account). Благодаря этому он ничего не впихнет в вашу папку, не сотрет ваши закладки и не насвинячит на вашем рабочем столе.

» sudo Метод, при котором обычному пользователю – обычно первому определяемому во время установки дистрибутива – временно предоставляются привилегии root для изменения конфигурации системы или установки программ. Графические версии – *gksudo* для Gnome и *kdesu* для KDE.



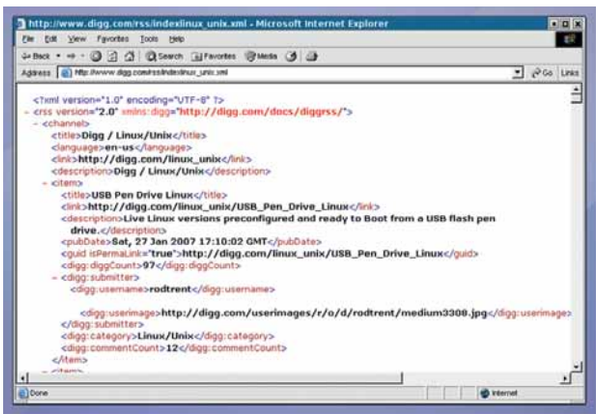
› С установкой Нет Доступа для 'Других', ваш домашний каталог будет невидим для визитеров.

» Через месяц Мы изучим науку резервирования данных в Linux.

Изучаем XSLT Превращаем RSS, фотоальбомы и многое другое, хранящееся в XML, в удобные web-страницы.

XSLT: Создадим

Мозги у вас на месте, а как с внешним видом? Умело оформив RSS, вы получите призы в обеих номинациях. А поможет вам в этом **Рейчел Проберт**.



› **Бедолаги-пользователи *Internet Explorer 6* – сделаем их убогий мир немного терпимее.**

Наш эксперт

Рейчел Проберт
Последние семь лет консультирует развивающиеся фирмы и создает для них web-приложения.

RSS – крутая технология, и ее популярность растет с бешеной скоростью. Многие из ваших любимых сайтов уже наверняка обзавелись новостными RSS-лентами, поскольку это упрощает и ускоряет распространение слова (или что там они пытаются распространять). Воспользоваться технологией RSS не отказался бы любой блоггер или сайтовладелец, но наскоком ею не овладеешь: итог – немало RSS-лент убогого вида, да еще и неудобочитаемых. Чаще всего встречаются два сценария:

» Хочу «одолжить» ленту новостей с чужого сайта, но оформить ее в собственном стиле.

» Не хочу, чтобы подписчики моей RSS-ленты видели в своих браузерах сырой XML, лучше создать нечто удобное для чтения.

Ответ на оба вопроса дает тема нашего занятия: XSLT.

Расширенный XML

Что же такое XSLT, и как мы будем им пользоваться? eXtensible Stylesheet Language Transformation или XSLT – это язык преобразования одних XML-документов в другие XML-документы. Он является подмножеством XSL, языка таблиц стилей XML. На данном уроке мы составим на XSLT таблицу стилей, «подгоняющую» существующую RSS-ленту к дизайну вашего сайта. Затем мы применим эту таблицу к исходному XML, чтобы пользователи могли просмотреть новости в человеко-читаемом виде.

Пользователям *Firefox 2* или *Internet Explorer 7* (фи) может показаться, что RSS-новости всегда выглядят неплохо. Дело в том, что новейшие браузеры воспроизводят RSS-ленты в удобочитаемом формате (независимо от наличия таблиц стилей в исходном коде), вместо бесхитростного отображения XML, как это было в недавнем прошлом. Но если у вас другой браузер (или версия), то при отсутствии таблицы стилей вы будете иметь несчастье созерцать «сырой» XML. Во врезке «*Браузеры и RSS*» на стр. 62 приводится список браузеров, умеющих отображать RSS в приличном виде.

Приступим

Наш проект можно разделить на четыре стадии. Мы возьмем RSS-ленту Linux с Digg.com, воспользуемся таблицей XSLT, преобразуем Digg

Образцы таблиц

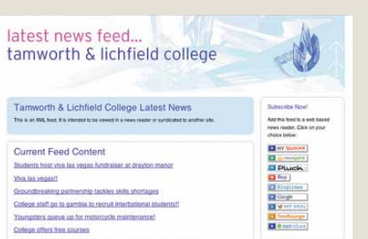
Есть несколько фантастических примеров стилевых таблиц, в которых можно черпать вдохновение. Вот наиболее известные адреса:

» http://newsrss.bbc.co.uk/rss/newsonline_uk_edition/front_page/rss.xml.

» http://feeds.feedburner.com/publisherbuzz.

Но эти ленты сделаны большими интернет-парнями. Пример, приведенный на иллюстрации, создан игроком поскромнее:

» www.tlc.ac.uk/xml/news-atom.xml.



› **Не надо громадного бюджета: просто освоим таблицы стилей!**

таблицу стилей

XML в элегантный HTML и отобразим его. Для урока вам понадобятся некоторые навыки и парочка полезных программ. Прежде всего, позаимствуем на www.digg.com ленту новостей о Linux/Unix. Чтобы усвоить урок, пригодится знакомство с XML, но честно говоря, оно не обязательно: если вы сумеете разобраться в содержании RSS-файла, проблем не должно быть. Однако прежде чем знакомиться с XSLT, необходимо понять XPath.

XPath для XML – примерно то же, что SQL для реляционных баз данных. Основная задача XPath – идентификация частей, или узлов, XML-документа. Рассмотрим следующий пример XML-файла:

```
<fridge>
  <fish species="trout" />
  <meat name="ermintrude" />
  <fish name="nemo" />
  <fish>wanda</fish>
</fridge>
```

XPath-эквивалент для выражения **SELECT fish FROM fridge** можно записать как /fridge/fish. Подошло бы и //fish, но этот вариант вернет всю рыбу, как в холодильнике, так и вне его. У XPath много общего с файловой системой Linux, поэтому знаток командной строки без труда разберется в навигации: ../ приведет к узлу, расположенному сразу над текущим (./).

Можно запрашивать XML не только по названию тэга, но и по атрибутам (*см. врезку «Выражения XPath», стр. 63*). Чтобы выбрать все компоненты холодильника, имеющие атрибут name (имя), можно воспользоваться выражением /fridge/*[@name]. Будут возвращены Nemo и Ermintrude, тогда как //fish[@*] вернет рыбу, имеющую хоть какой-то атрибут (всю, кроме Wanda). Можно указать //fish[@name="nemo"], тогда будет выбрана рыба, атрибут name которой равняется Nemo.

Итак, XPath использует «путевые» выражения для выбора узлов или групп узлов в XML-документе. Узлы выбираются, следуя указанному пути или шагам.

Для желающих получить более глубокие знания по этой теме, а заодно и попрактиковаться, у *Firefox* есть замечательный модуль под названием XPather (http://xpath.alephzarro.com). Установив XPather, можно загрузить в *Firefox* XML-файл, щелкнуть в окне правой кнопкой мыши и выбрать в контекстном меню пункт Show In XPather (Открыть в XPather). Работая со сложными XML-файлами, вы сможете проверять в XPather, соответствуют ли возвращаемые узлы вашим ожиданиям.

Попробуем реальные примеры. Загрузите www.digg.com/rss/indexlinux_unix.xml в *Firefox* и откройте исходный код страницы. Скопируйте его в файл на жестком диске, затем откройте этот локальный файл в *Firefox*. Мы делаем это потому, что последние версии *Firefox* конвертируют RSS в HTML. Локальные файлы не конвертируются, так что откроем окно XPather. Сперва попытаемся найти заголовок RSS-канала. В окне ввода XPather наберите **/rss/channel/title** и нажмите кнопку Eval (Вычислить). Если все прошло гладко, результатом будет следующее значение: «Digg/Linux/Unix». Теперь испробуем **//item/title** – должны возвратиться 40 значений, отображающих заголовки всех статей данной ленты.

Последний шаг перед началом действительно сложной работы – это установка *xsltproc*, инструмента командной строки для обработки XML-документов с помощью таблиц XSLT. Это часть C-библиотеки XSLT для Gnome, способная, впрочем, работать и в отрыве от этого рабочего стола. Если ваш дистрибутив использует *APT*, просто наберите приведенную ниже строку; если нет – обратитесь за инструкциями об установке ПО к менеджеру пакетов своего дистрибутива.

```
sudo apt-get install xsltproc
```

Охота за контентом

В конечном итоге, мы будем иметь дело с тремя взаимосвязанными файлами: исходным XML, XSLT-таблицей, которую мы собираемся создать, и выходным HTML, преобразованным *xsltproc* (*см.* **Рис. 1**, *ниже*). Первый шаг – определение RSS-ленты, которую мы стилизуем под свой сайт. Вы уже знаете, что я выбрала канал Linux/Unix Digg на www.digg.com/rss/indexlinux_unix.xml.

Потратьте некоторое время на изучение исходного XML-файла нашей ленты, это пригодится для создания XSLT-таблицы. Если ваш браузер воспроизводит ленты в читаемом формате, понадобится открыть исходный код выбором пункта контекстного меню View Page Source [Просмотр исходного кода страницы]. Этот-то файл нас и интересует. Наиболее важная для нас информация разделена на узлы: канал (channel), заголовок (title) и пункт (item). На данном этапе неплохо бы определить вид нашей страницы новостей. Подумайте, какие части вам хотелось бы включить и как они будут выглядеть.

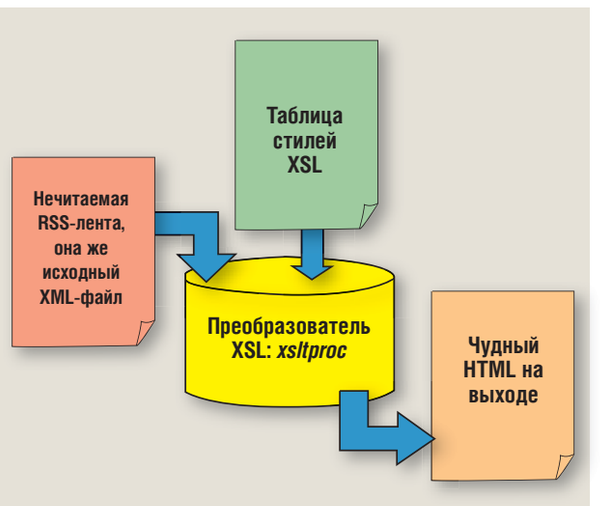
Мясо, картошка и лук

Держите копию исходного кода RSS-ленты открытой, чтобы можно было быстро взглянуть на него в случае необходимости. Теперь войдите в свою рабочую директорию и создайте файл с названием **digg.xsl**. Откройте этот файл в любым текстовом редакторе, оказавшемся под рукой. Я работаю в Kubuntu и пользуюсь *Kate*… ну, это неважно.

Начнем нашу таблицу так:

```
<xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
 xmlns:digg="http://digg.com/docs/diggrss/">
<xsl:output method="html"/>
<xsl:variable name="title" select="/rss/channel/title"/>
```

Таблица стилей XSLT начинается с элемента **<xsl:stylesheet>** и содержит операторы, управляющие воспроизведением. Мы должны также включить атрибут **version="1.0"**.



(Рис. 1) Волшебник *xsltproc* – ну как он это делает?

» Чтобы работать с элементами и атрибутами XSLT, в начале документа нужно объявить пространство имен XSLT (xmlns). Пространство имен XML – это метод разрешения конфликтов имен. В RSS есть предопределенные элементы, поэтому издатели вроде iTunes и Digg для удобства пользователей расширяют свои RSS-ленты дополнительными тэгами. Объявив пространство имен, мы получаем доступ к расширенным тэгам, образуя выражения объединением пространства имен с названием нового тэга через двоеточие (:), например digg:title. Тэг title входит в пространство имен digg, поэтому он не будет конфликтовать с тэгами, заранее определенными в XML-спецификации RSS. Мы начали с xmlns:xsl="http://www.w3.org/1999/XSL/Transform, что указывает на пространство имен W3C для XSLT.

В четвертой строке указан выходной формат html: ведь мы собрались построить HTML-файл! И последнее. Мы будем обращаться к заголовку канала неоднократно, поэтому, для экономии процессорного времени, установим переменную title, доступную в пределах всего нашего кода. Переменная title получает значение, возвращаемое выражение Xpath, заданное атрибутом select (/rss/channel/title). В нашем примере оно равняется Digg/Linux/Unix. Переменные доступны по нотации $title.

Итак, приправив наш документ лучком предписаний, начнем готовить мясо и картошку: элемент xsl:template (template – шаблон). Наберите

```
<xsl:template match="/">
```

Шаблон применяется ко всем узлам, отвечающим выражению Xpath, указанному в атрибуте match. Мы создаем HTML-документ, поэтому нам понадобится лишь один набор тэгов <html>. Следовательно, наш первый шаблон должен соответствовать только одному узлу. Можно было бы использовать любое выражение, возвращающее единственный узел, например "/rss" или "/rss/channel", но вернее будет "/", потому что такой узел грантированно один.

Пора задать разметку HTML-документа. Наберите следующее:

```
<html>
<head>
<title><xsl:value-of select="$title" /></title>
</head>
```

Как вы уже заметили, заголовок страницы задается переменной $title. Нотация, которую мы используем для этого – элемент XSLT <xsl:value-of>. Элемент value-of работает как с переменными, так и с выражениями XPath.

Скорая помощь

В мире Linux есть замечательное ПО для работы с XML. Взгляните на IDE Treebeard по адресу: http://treebeard.sourceforge.net/index.php.

Браузеры и RSS

Таблица показывает, как разные браузеры воспроизводят RSS, то есть какие не делают ничего и показывают исходный код XML, какие используют таблицу стилей ленты, а какие подавляют функции таблицы стилей собственным процессором. В настоящее время не существует браузера, который бы пользовался своим процессором лишь при отсутствии таблицы стилей. Если у вас *Internet Explorer 7* или *Firefox 2*, рекомендую дополнительно обзавестись одним из тех браузеров, которые «уважают» таблицы, не подавляя их.

Браузер	Бездействие	Таблица	Подавление
IE 7			✓
IE 6		✓	
pre-IE 6	✓		
Firefox 2			✓
Firefox 1.5		✓	
pre-Firefox 1.5	✓		
Opera 9		✓	
pre-Opera 9	✓		
Konqueror	✓		

Теперь добавьте

```
<body>
<div style="width:250px; float:right; padding:10px; margin:10px; border: thin dashed  lightgrey;">
К чему это все? Это RSS-лента с сайта Digg. Благодаря полученным навыкам работы с XSLT, новости с нее читаются и отображаются довольно мило.<hr />Если вы написали собственную RSS-ленту и предпочли бы включить ее вместо Diggs, можете использовать данный контейнер для описания целевой аудитории и способа подписаться на эту ленту.
</div>
```

Возможности стилизации шаблона поистине безграничны. Я поместила в контейнер пояснения о содержании RSS и придала документу некоторый стиль и правила поведения с помощью CSS. Тэг делит содержание страницы на логические блоки. Я хочу сказать, что определяет секции, которыми легко управлять, манипулировать, менять их стиль и которые легко читать. Обратите внимание на то, как я стилизовала контейнер , определив его положение, ширину, заполнение, границы и наличие рамки. Для творческого вдохновения, проанализируйте другие CSS.

Расставим по местам

Определив раскладку страницы, можно подумать о том, где применить шаблон. Попробуйте набрать следующее:

```
<h1>Welcome to today's <xsl:value-of select="$title" /> News</h1>
<xsl:apply-templates select="rss/channel/item"/>
<p>Пока это все, но мы скоро вернемся.</p>
</body>
</html>
</xsl:template>
```

Мы опять, как видите, воспользовались элементом <xsl:value-of>, чтобы поместить заголовок на страницу. Затем мы применили элемент <xsl:apply-templates> для вызова второго шаблона и добавили атрибут select: он будет обрабатывать лишь те узлы, значения которых соответствуют заданному выражению – в нашем случае это все пункты в канале RSS-ленты, т.е. статьи новостей. Фактически, мы встроили все пункты канала в соответствующую секцию HTML-кода (считайте это функцией отображения). Наконец, чтобы удостовериться в правильности размещения шаблона, мы добавили завершающую часть страницы.

Теперь необходимо задать разметку для узлов channel/item (статей Digg). Так как критерий match нашего шаблона возвращает группу узлов (несколько узлов, или статей), шаблон применяется к каждому узлу из этой группы.

```
<xsl:template match="rss/channel/item">
```

Первым делом нужно создать заголовок статьи, со ссылкой на основной ее текст (для тех, кто захочет прочесть ее целиком). Сделать это не так-то просто:

```
<h2>
<xsl:element name="a">
<xsl:attribute name="href"><xsl:value-of select="./link" /></xsl:attribute>
<xsl:attribute name="target">_new</xsl:attribute>
```

Выражение xsl:element служит для создания тэгов средствами XSLT. Уместен вопрос: почему бы не воспользоваться обычными гиперссылками HTML? Дело в том, что наши атрибуты href создаются динамически, а мы не можем поместить один XML-элемент в другой. Значит, нижеследующая строка просто не сработает:

```
<a href="<xsl:value-of select="./link" />" target="_new">
```

Элемент xsl:attribute срабатывает лишь тогда, когда он попадает внутрь xsl:element, и используется для добавления атрибутов к тэгу, который мы конструируем.

У XSLT есть еще один «туз в рукаве». В русле продуктовой темы, это будет наш десерт. Речь идет об элементе xsl:if, работающем с функцией contains(). В XSLT больше сотни встроенных функций, покрывающих почти все ваши нужды. Хотите быть круче – можете встроить расширения с помощью своего любимого языка программирования

(если процессор XSLT его поддерживает). Пользуясь xsl:if, давайте попробуем сделать вот что: если заголовок статьи Digg содержит слово «Linux», пусть он выводится красным цветом. Добавьте в код следующие строки:

```
<xsl:if test="contains(./title,'Linux')">
<xsl:attribute name="style">color:red;</xsl:attribute>
</xsl:if>
<xsl:value-of select="./title" />
</xsl:element>
</h2>
```

Теперь можно воспользоваться элементом <xsl:value-of> и выбрать интересующие нас узлы:

```
<strong>Published on: </strong><xsl:value-of select="./pubDate"/><br />
<strong>Number of Diggs: </strong><xsl:value-of select="./dig:diggCount" />
<br />
<strong>Number of comments: </strong><xsl:value-of select="./digg:commentCount" />
<p><xsl:value-of select="./description" /></p><hr />
```

Обратите внимание на метод обращения к расширенным тэгам Digg. Не забывайте закрывать тэги – таблица должна быть аккуратной. В конце добавьте

```
</xsl:template>
</xsl:stylesheet>
```

Из RSS в HTML через xsltproc

Настал момент истины! Таблица выглядит прекрасно, но работает ли она? Надеюсь, вы уже установили *xsltproc* и готовы запустить ее для обработки созданной таблицы и XML-файла Digg. Тогда откройте командную строку и вызовите *xsltproc*, указав название таблицы стилей, а за ним имя файла, к которому нужно применить таблицу. Если таблица стилей встроена в XML-документ с помощью соответствующей инструкции, название таблицы указывать незачем, *xsltproc* автоматически определит и использует ее. По умолчанию, результат выводится в stdout (стандартный вывод). Файл для вывода можно указать параметром **-o**.

Усвоили? Остается набрать следующее:

```
xsltproc -o /путь/к/output.htm /путь/к/digg.xsl http://www.digg.com/rss/indexlinux_unix.xml
```

Если все прошло гладко, дело сделано! Заходите в свою рабочую директорию и открывайте файл **output.htm**. Вы увидите там чудесно оформленную новостную ленту.

Этот простейший шаблон дает лишь понятие о том, как с помощью пары элементов и функций XSLT создать страницу новостей из «позаимствованной» RSS-ленты. Украшать страницу можно и дальше, особенно набравшись опыта работы с CSS. А сейчас пора интегрировать файл в ваш сайт и оформить его по собственному вкусу.

Процессоры XSLT существуют во многих языках программирования. Например, чтобы преобразовать XML-файл в ASP.NET с помощью Mono, можно сделать следующее:

```
<asp:xml DocumentSource="rss.xml" TransformSource="digg.xsl" runat="server" id="myNews" />
```

О подходящей альтернативе можно справиться в руководстве по языку.

Оформите свою RSS-ленту

Но это еще не все. Создав RSS-ленту (например, для своего блога), который отображается браузером в виде исходного XML, к ней можно добавить таблицу стилей и сделать ее читаемой. Чтобы применить таблицу стилей XSLT к своей ленте, нужно дополнить файл RSS. Введите следующую строку, она укажет браузеру расположение XSL-файла:

```
<?xml-stylesheet href="rss.xsl" type="text/xsl" media="screen"?>
```

Вот и вся необходимая модификация для данного файла.

Итак, мы сделали это! Взяв исходный XML-файл готовой ленты, мы преобразовали его с помощью таблицы XSLT, чтобы его можно было встроить в сайт. А еще мы воспользовались новообретенными навыками, чтобы присоединить стилевую таблицу к XML-файлу, подготовленному для распространения. Два в одном! Все большее количество браузеров получают поддержку RSS, и недалек тот день, когда необходимость в XSLT полностью отпадет. А пока этого не случилось, минимум усилий может существенно облегчить жизнь пользователям *Internet Explorer 6* и *Opera*.

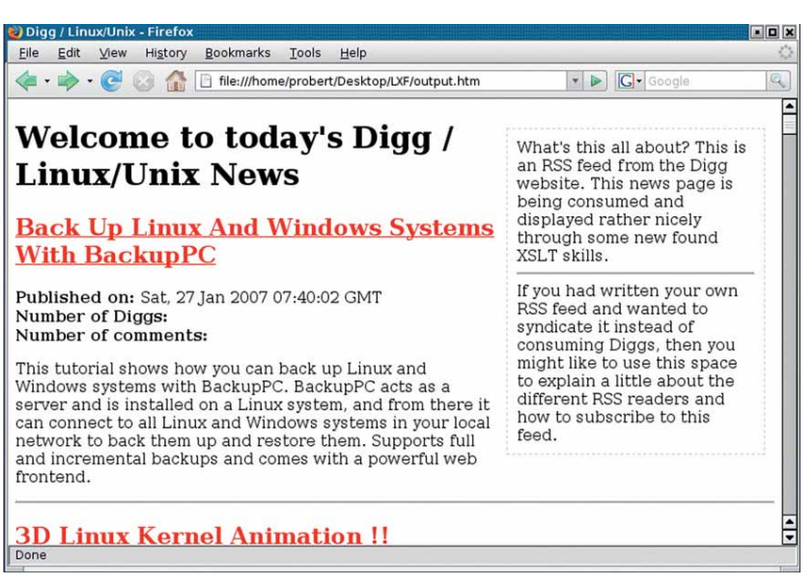
XSLT не привязан к RSS, его можно применить к любому XML-файлу. Например, *Picasa* от Google поддерживает экспорт фотоальбома в формате XML. С вашим новым опытом нетрудно преобразовать этот XML в Интернет-фотоальбом! LXF

Выражения XPath

Выражение	Описание
имяузла	Выбирает все дочерние узлы указанного узла.
/	Выбирает корневой узел.
//	Выбирает все узлы документа, соответствующие заданному критерию, независимо от их местонахождения.
.	Выбирает текущий узел.
..	Выбирает родительский узел текущего узла.
@	Выбирает атрибуты.

Скорая помощь

HTML-разметка в стилевой таблице XSL должна быть безупречной с точки зрения XML-форматирования. Это означает, что тэги нужно располагать в правильном порядке, обязательно соблюдать регистр названий тэгов, а значения атрибутов заключать в кавычки.



› Миссия выполнена. И не смейте больше огорчать взоры серферов.

SugarCRM: Как

ЧАСТЬ 2: Разбираясь с конфигурацией, **Марк Бэйн** покажет, как сломить упрямство *SugarCRM* и не сломать приложение (окончательно).

Наш эксперт

Марк Бэйн
Свою первую программу Марк написал еще для ZX81. Он много лет работает с Unix и Oracle, а также читает лекции и пишет книги. Его вторая книга, «Руководство разработчика *SugarCRM*» («The *SugarCRM* Developer's Manual»), скоро выйдет из печати.

Итак, вы уже установили *SugarCRM* на ваш чистенький сервер или, на худой конец, ознакомились с демо-версией этой системы в Интернете (*см. ссылку во врезке справа*). Думаю, теперь вы согласитесь, что она вполне готова удовлетворить все потребности стандартного пользователя.

Однако вы-то, наверняка, нестандартный пользователь! И наверняка программа выглядит не так или говорит с вами на разных диалектах. Да и вообще, хоть она и подходит для вашего бизнеса, но не совсем. А раз так, поговорим сегодня о том, как настроить *SugarCRM* в соответствии с потребностями пользователя.

Смотреть и видеть

Обратите внимание на выпадающий список с надписью «Тема» (Theme) в нижней части экрана *SugarCRM* в вашем браузере. С его помощью можно менять внешний вид программы, выбирая одну из тем: их довольно много, от стильной и профессиональной White Sands до совершенно несуразной Awesome 80s (впрочем, это мое личное мнение: бывают ведь и любители розового и зеленого текста на черном фоне).

Теперь можно перелицевать *SugarCRM*. Выбор не слишком впечатляет? Тогда создадим собственную тему. Не с нуля, конечно: возьмем одну из существующих и подправим ее.

Напомню, в прошлый раз мы загрузили архив со свежей версией *SugarCRM* (на сегодняшний день это 4.5.0h) с сайта программы и распаковали его содержимое в корень web-сервера (или в каталог по вашему вкусу), а потом заменили имя каталога **SugarOSFull-4.5.0h** на более благозвучное.

Откройте окно терминала, перейдите (командой *cd*) в каталог с *SugarCRM* и наберите **ls**. На экране появится содержимое директории: подкаталоги и файлы. В глаза сразу же бросается, что файлы все написаны на PHP (удивляться нечему: на PHP написана и сама *SugarCRM*). Но нам пока не до этого. Темы находятся в подкаталоге **themes**, вот его содержимое:

```
bainm@hector:/www/lxf > ls themes
```

Awesome80s	Links	Sugar
BoldMove	Love	SugarClassic
Default	Paradise	SugarLite
Default	Retro	Sunset
FinalFrontier	RipCurl	VintageSugar
GoldenGate	Shred	WhiteSands

Вы уже поняли: в каждом из подкаталогов – файлы соответствующей темы. Создадим еще одну.

Создать новую тему

Для начала скопируем папку с темой, взятой за основу для модификации, в новый каталог, например:

```
bainm@hector:/www/lxf/themes> cp -r SugarClassic lxf
```

Разберемся, какие файлы используются для создания темы:

```
bainm@hector:/www/lxf/themes> ls lxf
calendar-win2k-cold-1.css header.html menu.js
config.php header.php navigation.css
cookie.js images style.css
footer.php layout_utils.php
```

Самый главный из них – файл **config.php**. Именно он дает системе *SugarCRM* доступ к остальным файлам темы. В **config.php** много чего понаписано, но нам интересны две следующие строки:

```
$theme_name = “Sugar Classic”;
$theme_description = “Classic Sugar theme”;
```

Мы их отредактируем – заменим название темы и ее описание:

```
$theme_name = “Linux Format”;
$theme_description = “Linux Format theme”;
```

При следующем входе в систему вновь созданная тема *Linux Format* появится в списке тем. Пока она ничем не отличается от Classic *SugarCRM*, ведь мы изменили только имя. Пришла пора настоящих реформ!

Забраться в CSS

Теперь, когда мы создали новую тему и сообщили о ней программе, можно придать ей индивидуальный стиль. Для этого нужно кое-что поменять в файле… ясное дело, **style.css**.

Месяц назад Мы установили SugarCRM на LAMP-сервер.

это настроить

В нем (согласно его расширению – Cascading Style Sheet, каскадная таблица стилей) находится таблица стилей, задающая цвета, шрифты и прочее. Наш урок посвящен не таблицам стилей, и CSS как таковыми мы заниматься не будем: рассмотрим лишь небольшой пример, а в остальном вы разберетесь сами.

Рассмотрим раздел, в котором определяется, как отображать гиперссылки:

```
a:link, a:visited {
        color: #444444;
        font-size: 11px;
        text-decoration: underline;
}
a:hover {
        color: #666666;
        text-decoration: underline;
}
```

Все это означает, что гиперссылки – как те, что вы уже видели, так и непосещенные – отображаются одним и тем же серым цветом, причем при наведении курсора на ссылку оттенок серого меняется на более светлый; и все ссылки всегда подчеркнуты. Давайте внесем некоторое разнообразие: добавим после строки **text-decoration** в разделе a:hover такую строку:

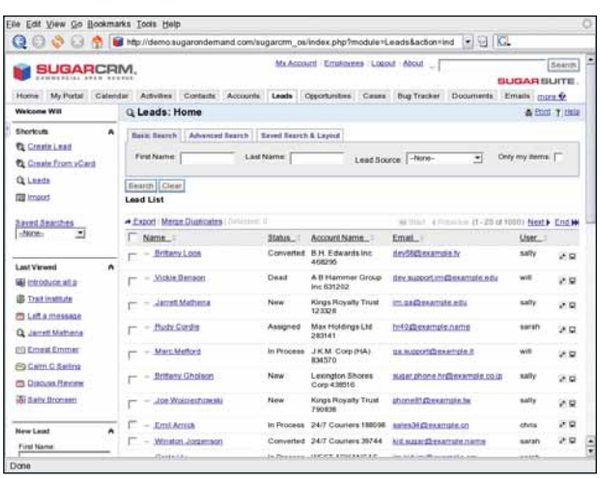
```
background-color:yellow; /*highlight the link*/
```

Чтобы увидеть результат этих изменений, не нужно входить в систему заново: просто обновите страничку. Теперь, при наведении курсора, фон гиперссылки меняется на желтый.

У тех, кто имел дело с таблицами стилей CSS, наверняка уже есть идеи по настройке внешнего вида темы на свой вкус. С остальными я поделюсь парой советов. Чтобы все ссылки отображались заглавными буквами, добавим строку в раздел **a:link**:

```
text-transform: uppercase;
```

Может быть, вы не стремитесь к супер-эффектам, а ограничитесь заменой основных цветов экрана. Тогда загляните в раздел **body**. Он предоставляет немало возможностей – например, позволяет указать цвет фона страницы:



› Тема White Sands выглядит вполне профессионально, но ее, как и другие темы, легко модифицировать.

```
background-color: yellow;
```

Немного поэкспериментировав, вы добьетесь своего идеала. Можете и Awesome 80s привести в приличный вид. Однако ваши пользователи больше заинтересованы в том, чтобы *SugarCRM* говорила с ними на понятном им языке; и вам по силам им помочь.

Адаптироваться к бизнесу

Взглянув на экран *SugarCRM*, вы приметите ряд вкладок. Некоторые из них пригодятся в любой организации: Home (Главная), Contacts (Контакты), Notes (Заметки), Meetings (Встречи), Calendar (Календарь) и Tasks (Задачи). Есть и другие вкладки, которые могут использоваться торговыми организациями для учета товарооборота: Opportunities (Возможности), Cases (Объекты) и Leads (Направления).

Хорошо, если внутри вашей организации используется та же терминология. А если нет? Приучать коллектив использовать стандартные термины *SugarCRM* вместо тех, что они впитали с молоком матери? Погодите смеяться: сам знаю, что это утопия. Переламывать себя сотрудникам незачем, гораздо проще кое-что переделать в *SugarCRM*.

Представим себе фирму проката автомобилей. Вместо Направлений у них New business (Новый бизнес), вместо вкладки Возможности они используют Inquiries (Запросы), а вместо вкладки Объекты – Bookings (Бронирования). На этом языке они изъяснялись последние 20 лет, и будут продолжать изъясняться еще 20 лет. Уж лучше поменять названия вкладок!

К счастью, ничего сложного в этом нет – едва ли вы не сумеете отредактировать пару PHP-скриптов, сейчас скажу каких. Но сначала знайте: лучше этого не делать! И вот почему: при установке новой версии программы эти файлы могут быть перезаписаны, и вам придется вносить изменения заново. Так или иначе, файл перед вами: **include/language/en_us.lang.php**. Найдите в нем раздел, содержащий

```
moduleList' =>
array (
'Home' => 'Home',
'Dashboard' => 'Dashboard',
'Contacts' => 'Contacts',
'Accounts' => 'Accounts',
'Opportunities' => 'Opportunities',
'Cases' => 'Cases',
'Notes' => 'Notes',
'Calls' => 'Calls',
'Emails' => 'Emails',
'Meetings' => 'Meetings',
'Tasks' => 'Tasks',
'Calendar' => 'Calendar',
'Leads' => 'Leads',
```

Этот массив представляет собой список вкладок. Как явствует из его имени, вкладки в *SugarCRM* называются «модулями», поскольку каждая реализуется отдельным модулем, но об этом чуть позже. Вы, поди, недоумеваете: как изменить названия вкладок, если нельзя редактировать файл, в котором они хранятся? Легко: создадим специальный файл, содержащий обновленные названия вкладок.

Обратите внимание на папку **custom** внутри каталога *SugarCRM*: в ней мы и займемся языковедением. Как вы уже сообразили, нам нужен файл **custom/include/language/en_us.lang.php**. »

Скорая помощь

Если SugarCRM у вас не установлена, можете изучить демо-версию, расположенную по адресу **http://demo.sugarondemand.com/sugarcrm_os/index.php?action=Login&module=Users**.

Я говорил вам, что этот урок – не про CSS; добавлю, что он и не про PHP. Достаточно сказать, что создается двумерный массив, он содержит названия вкладок, и в нем можно переименовывать существующие вкладки и добавлять новые. Для переименования трех вышеупомянутых вкладок достаточно добавить в этот файл следующие строки:

```
$app_list_strings['moduleList']['Leads'] = 'New Business';
$app_list_strings['moduleList']['Opportunities'] = 'Inquiries';
$app_list_strings['moduleList']['Cases'] = 'Bookings';
```

Осталось только сохранить его и обновить страничку в браузере. На месте трех старых вкладок появились новые – New Business, Inquiries и Bookings.

Вроде неплохо, но откройте любую из этих вкладок – и увидите, что содержимое-то осталось прежним. Например, на вкладке New Business (бывшей Leads) остались ссылки 'Create Lead', 'List Leads' и 'New Lead'. Тут ваши сотрудники, пожалуй, окончательно запутаются! Значит, работа по настройке не окончена.

Мы уже знаем, что названия вкладок для текущего языка хранятся в файле **include/language/en_us.lang.php**, и изменить их можно через файл **custom/include/language/en_us.lang.php**, прописав в нем новые названия вкладок. Аналогично, каждая вкладка-модуль тоже имеет собственный языковой файл.

Вернемся в папку *SugarCRM* и найдем каталог **modules**. Именно в нем хранятся модули:

```
bainm@hector:/www/lxf> ls modules
Accounts
ACL
ACLActions
ACLRoles
Activities
...
```

и т.д. Модуль **Leads** расположен в одноименном каталоге. Как и остальные модули, реализующие вкладки, этот модуль содержит свои языковые настройки в папке **language**:

```
bainm@hector:/www/lxf> ls modules/Leads/language
en_us.help.DetailView.html en_us.help.index.html
en_us.help.EditView.html en_us.lang.php
```

Ну, вы поняли: файл **en_us.lang.php** содержит определения строковых констант для текущего языка.

```
bainm@hector:/www/lxf> grep -i lead modules/Leads/language/
en_us.lang.phpl head -10
'ERR_DELETE_RECORD' => 'en_us A record number must be specified
to delete the lead.',
'LBL_BACKTOLEADS' => 'Back To Leads',
'LBL_BUSINESSCARD' => 'Convert Lead',
'LBL_CONTACT_INFORMATION' => 'Lead Information',
'LBL_CONTACT_NAME' => 'Lead Name:',
'LBL_CONTACT_OPP_FORM_TITLE' => 'Lead-Opportunity:',
'LBL_CONTACT' => 'Lead:',
'LBL_CONVERTLEAD_BUTTON_KEY' => 'V',
'LBL_CONVERTLEAD_TITLE' => 'Convert Lead [Alt+V]',
'LBL_CONVERTLEAD' => 'Convert Lead',
```

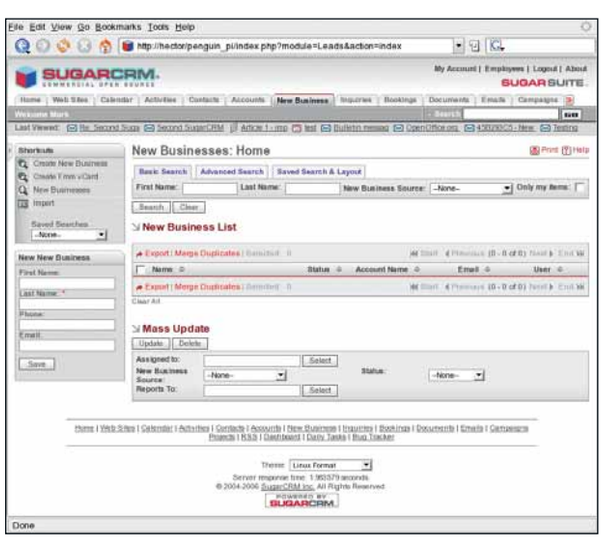
Как и основной файл языка системы, его редактировать не следует. Есть даже строки, менять которые категорически не рекомендуется:

```
// НЕ ПОРТИТЕ ЭТИ СТРОКИ: ОНИ ЗАДАЮТ СООТВЕТСТВИЯ
'db_last_name' => 'LBL_LIST_LAST_NAME',
'db_first_name' => 'LBL_LIST_FIRST_NAME',
'db_title' => 'LBL_LIST_TITLE',
'db_email1' => 'LBL_LIST_EMAIL_ADDRESS',
'db_account_name' => 'LBL_LIST_ACCOUNT_NAME',
'db_email2' => 'LBL_LIST_EMAIL_ADDRESS',
//КОНЕЦ ЗАПРЕТНЫХ СТРОК
```

Снова создадим файл **custom/modules/Leads/language/en_us.lang.php** с языковыми настройками.

```
<?php
$lead_single = 'New Business';
$lead_title = $lead_single . 'es';

$mod_strings['LBL_BACKTOLEADS'] = 'Back To ' . $lead_title;
$mod_strings['LBL_DEFAULT_SUBPANEL_TITLE'] = $lead_title;
$mod_strings['LBL_DUPLICATE'] = 'Similar ' . $lead_title;
$mod_strings['LBL_LIST_MY_LEADS'] = 'My ' . $lead_title;
$mod_strings['LBL_MODULE_NAME'] = $lead_title;
$mod_strings['LBL_MODULE_TITLE'] = $lead_title . ': Home';
$mod_strings['LBL_SELECT_CHECKED_BUTTON_LABEL'] = 'Select
Checked ' . $lead_title;
$mod_strings['LBL_SELECT_CHECKED_BUTTON_TITLE'] = 'Select
Checked ' . $lead_title;
$mod_strings['LNK_LEAD_LIST'] = $lead_title;
$mod_strings['LBL_BUSINESSCARD'] = 'Convert ' . $lead_single;
$mod_strings['LBL_CONTACT_INFORMATION'] = $lead_single .
'Information';
$mod_strings['LBL_CONTACT_NAME'] = $lead_single . ' Name:';
$mod_strings['LBL_CONTACT_OPP_FORM_TITLE'] = $lead_single . '-
Opportunity:';
$mod_strings['LBL_CONTACT'] = $lead_single . ':';
$mod_strings['LBL_CONVERTLEAD_TITLE'] = 'Convert ' .
$lead_single . ' [Alt+V]';
$mod_strings['LBL_CONVERTLEAD'] = 'Convert ' . $lead_single;
$mod_strings['LBL_LEAD_SOURCE_DESCRIPTION'] = $lead_single .
'Source Description:';
$mod_strings['LBL_LEAD_SOURCE'] = $lead_single . ' Source:';
$mod_strings['LBL_LIST_CONTACT_NAME'] = $lead_single . 'Name';
$mod_strings['LBL_VIEW_FORM_TITLE'] = $lead_single . 'View';
$mod_strings['LBL_LIST_FORM_TITLE'] = $lead_single . ' List';
$mod_strings['LBL_LIST_LEAD_SOURCE_DESCRIPTION'] = $lead_
single . ' Source Description';
$mod_strings['LBL_LIST_LEAD_SOURCE'] = $lead_single . 'Source';
$mod_strings['LBL_NEW_FORM_TITLE'] = 'New ' . $lead_single;
$mod_strings['LBL_SEARCH_FORM_TITLE'] = $lead_single . 'Search';
$mod_strings['LNK_NEW_LEAD'] = 'Create ' . $lead_single;
?>
```

Сохраните этот файл и обновите текущую страницу в браузере. Но это еще не все. Нужно проверить, не содержат ли ссылок на этот модуль остальные модули. Например, в панели Mass Update на вкладке Opportunities (или Inquiries, если вы уже успели ее переименовать) остался текст 'Lead Source:'. Беглый поиск в файле **modules/Opportunities/language/en_us.lang.php** выявит переменную, которую надо поместить в файл **custom/modules/Opportunities/language/en_us.lang.php**, чтобы поправить дело:



Победа! Не так уж сложно заставить вашу версию *SugarCRM* говорить на одном языке с вашими пользователями.

Скорая помощь

Чтобы изменить логотип SugarCRM в левой части экрана, зайдите в систему как администратор и выберите пункт System Settings (Настройки системы). Новое изображение должно быть размером 212x40 пикселей и иметь прозрачный фон.

```
bainm@hector:/www/lxf> grep “Lead Source:” modules/Opportunities/
language/en_us.lang.php
‘LBL_LEAD_SOURCE’ => ‘Lead Source:’,
```

Таким образом, в файле **n_us.lang.php** для Opportunities нужно добавить следующую строку:

```
$mod_strings[‘LBL_LEAD_SOURCE’] = ‘New Business Source’;
```

Конечно, вручную менять значения переменных, раскиданных по разным файлам – не лучший способ, особенно если на презентации вашего продукта вам вдруг заявят: «Мы тут прикинули – вместо ‘New Business’ лучше написать ‘Fresh Business’».

Гораздо удобнее разместить все переменные в одном файле, например, **custom/include/lxf_variables.php**, а затем включать его, где понадобится. Ваши переменные могут выглядеть так:

```
<?php
$lead_single = ‘New Business’;
$lead_title = $lead_single . ‘es’;
?>
```

а в файле **custom/modules/Opportunities/language/en_us.lang.php** тогда появится следующий фрагмент кода:

```
<?php
require_once(‘custom/ include/lxf_variables.php’);
$mod_strings[‘LBL_LEAD_SOURCE’] = $lead_single . ‘ Source’;
?>
```

Но я предупреждал: данный урок – не про PHP.

Озаглавить окно браузера

Под конец урока займемся заголовком окна браузера. Сейчас он такой: “*SugarCRM* – Commercial Open Source CRM”. Это как-то слабо связано с нашим проектом. Чтобы изменить его, добавим в файл **custom/include/language/en_us.lang.php** строку, меняющую значение переменной **LBL_BROWSER_TITLE**:

```
$app_strings[‘LBL_BROWSER_TITLE’] = ‘LXF – Customizing SugarCRM’;
```

Обычным манером, сохраните этот файл и обновите страничку – заголовок окна браузера изменится.

Менять, все менять!

Наше маленькое исследование *SugarCRM* показало: нет ничего такого, что нельзя было бы переделать по своему вкусу. Все приложение – лишь набор скриптов PHP, взаимодействующих с базой данных. Мы увидели, как легко создать (и модифицировать) новую тему. Меняли мы только цвет и стили оформления, но в каждой папке с темой есть каталог **images**, содержащий применяемые этой темой изображения, и пользователь может заменить их на свои собственные (*см. врезку слева*), и даже удалить все темы, кроме собственной (это пригодится, например, при продвижении вашего брэнда).

А как там с ограничениями? Просто упражняться с *SugarCRM* можно безгранично: ведь это же свободное ПО (разве что, как полагается, не удаляйте и не модифицируйте файлы, не предусмотрев резервной копии).

Зато для применения программы в коммерческой деятельности нужно учитывать положения лицензии *SugarCRM* Public License (SPL), это вариант лицензии Mozilla Public License, версия 1.1. Там указано, что можно менять, а что нельзя. Исходный код открыт, и менять можно абсолютно все, но:

1. Измененные версии файлов должны быть общедоступны.
2. Измененная программа должна сопровождаться документацией, описывающей изменения.
3. К программе обязательно должна прилагаться копия лицензии SPL.
4. Не взывайте к фирме SugarCRM Inc, если их новый релиз будет несовместим с вашей версией.

И еще кое-что: согласно лицензии SPL, нужно обязательно сохранять логотип ‘Powered by SugarCRM’ и информацию об авторских правах в нижней части страницы. А в остальном – все ваше.

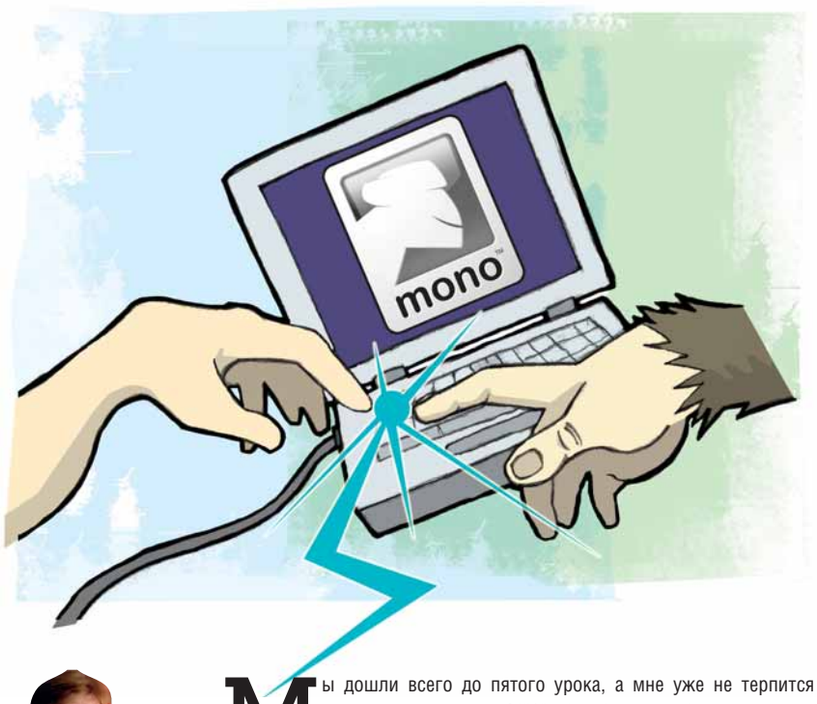
Полный текст лицензии можно найти на **www.sugarcrm.com/crm/SPL**.

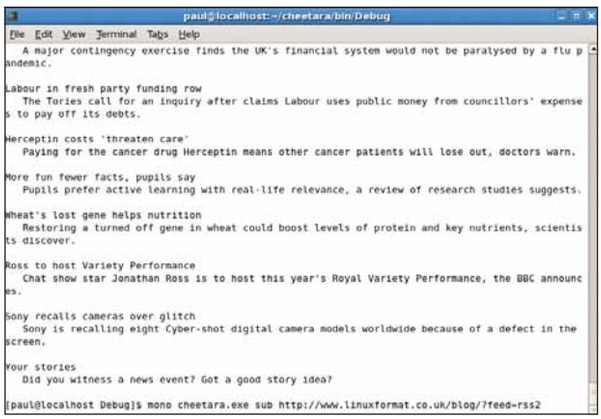
На сегодня все. Надеюсь, я сумел показать, как легко настраивается *SugarCRM*, даже если ваши знания PHP минимальны. Итак, за настройку! LXF

» **Через месяц** Зароемся в SugarCRM поглубже и добавим свои поля и вкладки.

Mono: Создаем

При вашем знании Mono обычные текстовые программы писать уже скучно. **Пол Хадсон** поможет разобраться с новым проектом, использующим графический интерфейс.





Возьмем немного кода от RSS-агрегатора из LXF89, *Cheetara*, и для красоты кое-что добавим.

Наш эксперт

Пол Хадсон
Пол Хадсон полагает, что Mono – лучшая вещь со времен мультфильма *Pinky and the Brain*, и сейчас поддерживает два проекта на основе Mono на SourceForge.

Мы дошли всего до пятого урока, а мне уже не терпится показать вам магию *GTK#*... Вообще-то я бы и погодил с показом, но столько читателей просили меня об этом – не игнорировать же народные массы! Надеюсь, вам понравились эксперименты с Mono – ведь это очень весело. Но, по моему мнению, стоит влезть в библиотеку *GTK*, как начнутся первые проблемы. Видите ли, *GTK* писали люди, не любившие принципы объектно-ориентированного программирования, но попытавшиеся реализовать их на C. Да еще он безнадежно усложнен и почти непознаваем для новичка. Но вам повезло: я рядом!

На первых четырех уроках вы создали программы Hello World, утилиту для поиска файлов, RSS-агрегатор и простой интерфейс для *Beagle*. На этом уроке, чтобы целиком сконцентрироваться на *GTK*, мы возьмем готовый код из RSS-проекта и создадим графический RSS-агрегатор, который я назову *Chomp*. *GTK* не столько сложен, сколько муторен – если вы что-то не вполне поняли, вернитесь назад на пару абзацев: нить рассуждений потерять очень легко.

Удар от Stetic

Как я уже сказал, *GTK* написан на C, но к нему существует интерфейс Mono/C#, вполне предсказуемо названный *GTK#*. Это очень тонкая обертка вокруг кода C, а значит, работать с ней довольно нудно. Но одну часть нудной никак не назовешь: это *Stetic* – инструмент для построения графических интерфейсов пользователя в стиле Drag&Drop, входящий в состав *MonoDevelop*. Если вы помните первый урок данной серии, мой дистрибутив – Fedora Core 6, и те, кто тоже его установил, уже могут использовать *MonoDevelop* со *Stetic*. Если у вас другой дистрибутив, убедитесь, что *MonoDevelop* в нем версии 0.12 или старше.

Первым делом нарисуем наш интерфейс с помощью *Stetic*. Никакой функциональности это не даст, зато даст представление о том, как будет выглядеть готовое приложение, а я всегда считал, что это позволяет шагнуть далеко вперед. Другая причина, почему стоит сначала спроектировать интерфейс – вы сразу же увидите, будет ли ваш проект неуклюж или невразумителен, и, в случае чего, заранее примете меры по редизайну интерфейса. Помните, что у лучших проектов интерфейс пользователя разрабатывается очень тщательно – не зацикливайтесь исключительно на написании кода в надежде, что GUI сложится само собой!

Прежде чем приступать, полностью обновите версию Fedora, потому что *Stetic* – сравнительно молодой инструмент, и лучше использовать свежую версию *MonoDevelop* для минимизации числа возможных отказов. Да почаще сохраняйте результаты, поскольку *Stetic* не страдает излишней стабильностью.

Итак, аккуратно выполните инструкции по работе в *MonoDevelop*, приведенные на следующей странице и продолжите чтение на стр. 70.

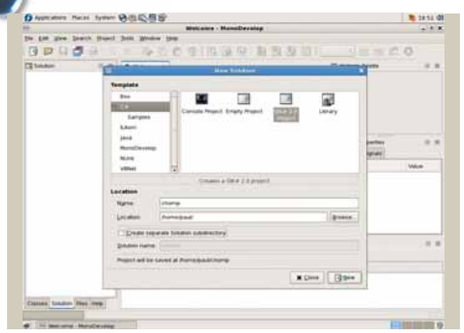
Stetic против Glade

Stetic и *Glade* очень похожи: оба поддерживают drag&drop для создания пользовательских интерфейсов на *GTK*. Но *Stetic* интегрирован в *MonoDevelop*, то есть может генерировать код, связывая виджеты с переменными. При желании можно использовать в Mono и *Glade*, но рекомендуется все-таки *Stetic*.

Месяц назад Мы использовали библиотеки *Beagle* для супер-быстрой разработки утилиты поиска.

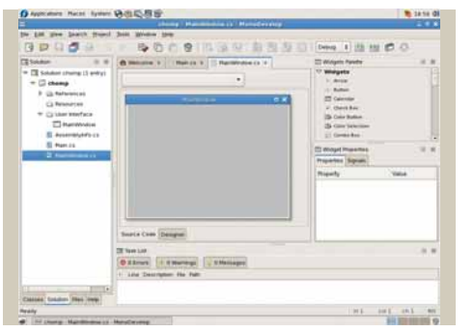
GTK-приложение

Часть 1 Создание интерфейса пользователя



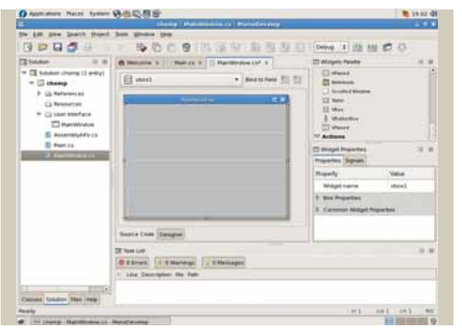
1 Создайте новый проект

Выберите проект C# из области шаблонов, а затем – Gtk# 2.0 Project, справа. Назовите его chomp, снимите галочку с Create Separate Solution Subdirectory и нажмите New.



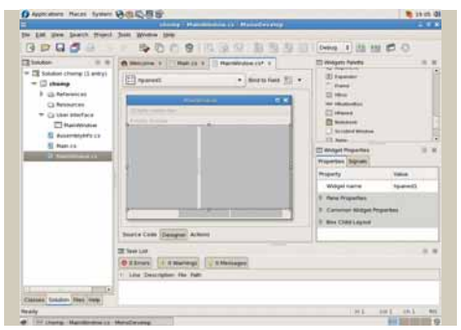
2 Откройте MainWindow

Откройте дерево пользовательского интерфейса в панели Solution и дважды щелкните на MainWindow. Откроется MainWindow.cs в режиме Designer.



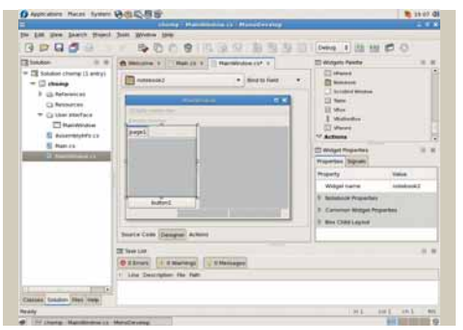
3 Добавьте четыре области

В палитре Widgets пройдите до Containers и перетащите VBox на окно. Щелкните на нем правой кнопкой и выберите Vbox1 > Insert Before для четырех областей.



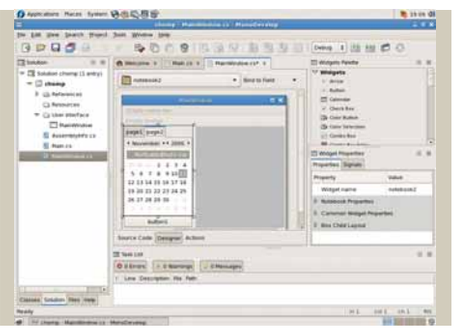
4 Преобразуйте области

Перетащите главное меню, панель инструментов, Hpaned и строку состояния во все четыре области, начиная с верхней. Разделите Hpaned на две примерно равные части, как показано на рисунке.



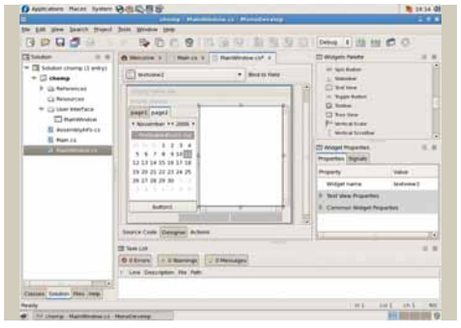
5 Новый Notebook

Перетащите VBox в левую часть Hpaned. Нажмите на ней и выберите Delete. Перетащите Notebook и Button на верхнюю и нижнюю стороны VBox.



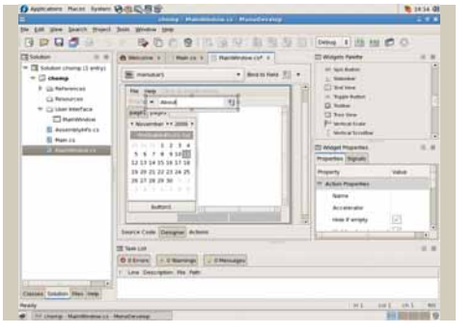
6 TreeView и Calendar

Перетащите Tree View на вкладку Page1 компонента Notebook. Щелкните на нем правой кнопкой, выберите Notebook1, затем Insert Page After. На вкладке Page2 вставьте Calendar.



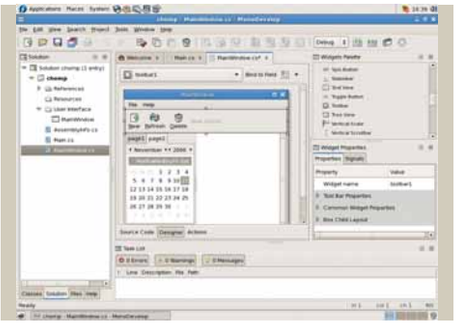
7 Добавьте полосы прокрутки

В большом пространстве справа от элемента Hpaned вставьте Scrolled Window, затем поместите в него Text View. Пользователи смогут прокручивать текст.



8 Организуйте меню

Дважды щелкните на пустом меню, наберите /'File', затем нажмите Enter. Таким образом, вы создадите подменю – наберите в нем 'Quit'. Создайте меню Help с подменю About.



9 Кнопки панели инструментов

Дважды щелкните по панели инструментов, выберите Select Icon, затем gtk-new. Добавьте две другие кнопки: одну с gtk-refresh, другую с gtk-delete. »

Часть 2 Реализуем функционал программы

Скорая помощь

Проверьте, что файл sitelist.txt содержит адреса URL лент, по одному на строке, и что он находится там же, где и файл chomp.exe. Обычно это каталог bin/Debug или bin/Release, если вы сделали релиз-версию.

Создание интерфейса позади, но для готового GUI еще кое-чего нехватает. Нужно сделать три вещи. Во-первых, назначить нормальные имена виджетам: пока что они называются button1, textview2 и т.д. Во-вторых, связать виджеты с полями: тогда мы сможем обращаться к ним из кода на C#. И в-третьих, изменить свойства виджетов, чтобы они выглядели как надо и делали то, что мы хотим.

Имя виджета – это просто одно из свойств, поэтому задачи 1 и 3 выполняются через панель Widget Properties, находящейся в правом нижнем углу окна *MonoDevelop*. Вы увидите кнопку Bind To Field в верху окна, которое только создали – я вам скажу, когда она понадобится.

Вот список требуемых изменений:

- Нажмите на кнопку New на панели инструментов toolbar (не на самой панели!) и измените ее имя на btnNew.
- Измените имя кнопки Refresh на btnRefresh.
- Измените имя кнопки Delete на btnDelete.
- Измените имя Tree View на tvFeeds. Нажмите Bind To Field.
- Снимите галочку с Show Day Names для Calendar (это его немного уменьшит). Нажмите Bind To Field.
- Измените имя Text View на txtFeed. Нажмите Bind To Field. Измените Hight Request на 300, а Width Request на 400 (теперь Text View не будет занимать весь экран).
- Измените имя кнопки под компонентом notebook на btnRefreshAll.
- Измените текст для страниц компонента notebook на By Feed для вкладки с Tree View и на By Date для вкладки с Calendar.

Добавим реализацию

Вот и все – с интерфейсом пользователя покончено, можно сфокусироваться на реализации методов. Наберем кое-какой код прямо в файле **MainWindow.cs**, а также установим обработчики сигналов для **MainWindow.cs** в *Stetic*. Сигналы – это внутренние *GTK*-сообщения, которые посылаются, когда что-нибудь происходит: например, пользователь нажимает кнопку или клавишу. По умолчанию, большая часть сигналов ничего не делает, но мы можем на них подписаться, назначив каждому обработчик с помощью *Stetic* – он даже создаст за нас базовый метод!

Прежде всего обработаем событие, возникающее, когда пользователь выделяет что-то в Tree View нашей программы. Этот компонент будет использоваться для отображения списка RSS-лент, на которые подписан пользователь, поэтому при выборе пользователем ленты из списка *Chomp* надо скачать последнюю версию RSS-файла и отобразить его в Text View справа. Щелчки пользователя можно обработать, отлавливая сигнал RowActivated – щелкните на Tree View, затем на панели Widget Properties перейдите на вкладку Signals. Найдите сигнал RowActivated и дважды щелкните там, где написано Click Here To Add A New Handler. Назовите его OnRowActivated и нажмите Enter. Щелкнув на кнопке Source Code, расположенной ниже окна дизайна формы, вы увидите, что *Stetic* создал следующий метод:

```
protected virtual void OnRowActivated(object o, Gtk.
RowActivatedArgs args)
{
}
```

Скорая помощь

GTK предусматривает масштабирование компонентов интерфейса при изменении размера окна, а также их сжатие и растяжение при переводе надписей на другие языки, смене шрифта и других параметров среды выполнения. Вот почему мы используем VBox и HBox, а не размещаем компоненты вручную.

Вроде неплохо, но тут вы и подошли к первому недостатку *GTK*. А именно: получить данные из Tree View гораздо сложнее, чем вы думаете!

Chomp требуется работать с лентами двумя способами. Если мы находимся на вкладке By Feed, то двойной щелчок на одной из лент в Tree View должен загрузить эту самую ленту. Если мы находимся на вкладке By Date, при двойном щелчке на дате в календаре следует загрузить все ленты, но показать новости только за указанную дату. Чтобы как-то унифицировать наш код, создадим метод ReadFeed(), который загружает URL, а затем фильтрует их по дате. При загрузке всех записей данной ленты мы передадим специальный аргумент DateTime.MinValue, обозначающий «игнорировать дату».

Вот код, который надо поместить в OnRowActivated:

```
TreeSelection select = tvFeeds.Selection;
TreeIter iter;
TreeModel model;
select.GetSelected(out model, out iter);
string val = (string)model.GetValue(iter, 0);
txtFeed.Buffer.Clear();
ReadFeed(val, DateTime.MinValue);
```

Первые пять строк иллюстрируют мои слова о занудстве *GTK*: таким манером мы извлекаем значения из компонента Tree View. Весьма запутанно, но зато годится для любого возможного способа использования Tree View.

Когда выбранное значение попадет в переменную val, мы очищаем буфер Text View и вызываем ReadFeed(), чтобы загрузить и отобразить RSS. Но давайте сначала разберемся с календарем. Для этого найдите сигнал DaySelectedDoubleClick и назначьте ему обработчик DayClicked. В режиме Source Code среды *MonoDevelop* должен появиться пустой метод DayClicked(), который выглядит так:

```
protected virtual void DayClicked(object sender, System.
EventArgs e)
{
}
```

Этот метод отлавливает двойные щелчки на календаре, а получить выбранную дату из компонента Calendar очень просто.

Но не спешите хвалить *GTK*: вот я вам сейчас покажу, как перебрать все элементы в Tree View. В DayClicked() надо поместить следующий код:

```
txtFeed.Buffer.Clear();
tvFeeds.Model.Foreach(FeedByDate);
```

Здесь мы очищаем текстовый буфер для входящих лент, а затем вызываем метод Foreach() модели данных Tree View. Метод FeedByDate() (его я еще не показывал – не гоните коней!) вызывается для каждого элемента в Tree View. А так как метод FeedByDate() вызывается методом Foreach(), он должен соответствовать конкретной функции-прототипу, то есть принимать определенный список параметров. Иначе он работать не будет.

Мы хотим, чтобы функция FeedByDate() читала каждую ленту, которую ей передают, затем посылала ее в ReadFeed() вместе с датой, определенной на компоненте Calendar. Это довольно просто:

```
bool FeedByDate(TreeModel model, TreePath path, TreeIter iter)
{
string url = (string)model.GetValue(iter, 0);
ReadFeed(url, new DateTime(calendar.Year, calendar.Month + 1,
calendar.Day));
return false;
}
```

Первая строка выцарапывает URL ленты из Tree View, после чего он передается методу ReadFeed() вместе с годом, месяцем и днем, выбранными в календаре. Но вы заметили calendar.Month + 1? *GTK* ведет отсчет месяцев с нуля. Зато дни и года он считает с 1. Я тоже не знаю, почему!

Пару выпусков назад мы рассматривали код, необходимый для чтения RSS-лент. На сей раз мы его используем, но с небольшими изменениями:

- Вывод будет происходить в текстовый буфер, а не в консоль.
- GUID'ы кэшироваться не будут – мы хотим, чтобы *Chomp* загружала все записи каждой ленты, а не записи, которые мы не видели.
- Если для времени указано DateTime.MinValue, имеется в виду «по дате фильтровать не надо».
- Иначе, получить дату публикации каждой новости и преобразовать ее в объект DateTime.

» Затем сравнить каждый объект DateTime с указанной датой и вывести только те записи, которые ей соответствуют.

Тут есть мелкий просчет: одни ленты, включая *Linux Format*, предоставляют дату публикации в формате Wed, 24 Jan 2007 11:02:43 +0000, а другие вместо '+0000' ставят 'GMT'. .NET понимает оба формата, но в Mono – возможно, по недосмотру – поддерживается только последний. В качестве обходного пути мы будем обрезать всю подстроку, начиная с символа +.

Время писать код

Вот и код. Если вы читали LXF89, то должны узнать его части!

```
protected virtual void ReadFeed(string feed, DateTime filter) {
  XmlDocument doc = new XmlDocument();
  doc.Load(feed);
  TextBuffer text = txtFeed.Buffer;
  XmlNodeList items = doc.SelectNodes("//item");
  foreach (XmlNode item in items) {
    if (filter == DateTime.MinValue) {
      text.Text += (item.SelectSingleNode("title").InnerText) + "\n";
      text.Text += (" " +  item.SelectSingleNode("description").
InnerText) + "\n\n";
    } else {
      string time = item.SelectSingleNode("pubDate").InnerText;
      if (time.Contains("+")) time = time.Substring(0, time.
 IndexOf("+"));
      DateTime thisdate = DateTime.Parse(time);
      if (filter.Day == thisdate.Day && filter.Month == thisdate.Month
 && filter.Year == thisdate.Year) {
        text.Text += (item.SelectSingleNode("title").InnerText) + "\n";
       text.Text += (" " + item.SelectSingleNode("description").
InnerText) + "\n\n";
      }
    }
  }
}
```

Две строки могут вызвать затруднение:

```
if (time.Contains("+")) time = time.Substring(0, time.IndexOf("+"));
DateTime thisdate = DateTime.Parse(time);
```

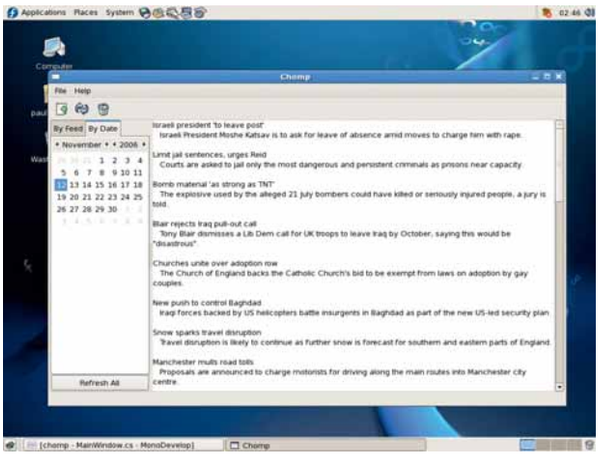
Первая означает «если строка содержит +, просьба записать в переменную time все от начала строки (от символа с номером ноль) до знака +»: таким образом отсекается часть +0000. Вторая строка означает «возьми строку time и преобразуй ее в тип DateTime.» Хранить дату как DateTime удобнее, потому что можно работать с днем, месяцем и годом как с числами, а не разбирать строку вручную.

Финишная прямая

Первая версия *Chomp* почти завершена, но в ней кое-чего не хватает. Двойной щелчок по лентам загружает все ленты. Двойной щелчок по дате загружает все ленты и фильтрует записи по выбранной дате. Но в какой момент мы загружаем ленты в программу? Ответ: а ни в какой!

Наш предыдущий XML-считыватель тоже загружал список лент, поэтому можно просто взять и модифицировать его для нашего интерфейса на *GTK*. А именно, давайте поместим каждый URL в Tree View, чтобы пользователи могли нажимать на них. Здесь опять проявится занудство *GTK*, поэтому возьмите себя в руки:

```
string[] sitelist;
if (File.Exists("sitelist.txt")) {
  sitelist = File.ReadAllLines("sitelist.txt");
} else {
  sitelist = new string[0];
}
// это значит, что мы хотим сохранить строку в строке Tree View, но в
действительности нам  нужен только URL!
```



› Результат: RSS-агрегатор на базе *GTK*, фильтрует по дате и по URL. Вопрос: куда двигаться дальше?

```
TreeStore store = new TreeStore (typeof(string), typeof(string));
foreach(string site in sitelist) {
  //  "Foo" это место размещения
  store.AppendValues(site, "Foo");
}
tvFeeds.Model = store;
tvFeeds.AppendColumn("URLs", new CellRendererText(), "text",0);
```

Чтобы показать данные, Tree View нужен по крайней мере один столбец. Смело отключайте заголовки в свойствах компонента *Stetic*, в них нет нужды. Теперь создайте файл **sitelist.txt** в вашем рабочем каталоге программы (наподобие **/путь/до/проекта/bin/Debug**), заполните его URL-адресами лент, и пусть поработает!

Весь код этого учебника, даже с некоторыми дополнительными возможностями *(см. ниже врезку Chomp 0.2)*, включен на диск. Но чтобы проект созрел для помещения на SourceForge, не мешает его усовершенствовать. Оставляю *Chomp* в ваших умелых руках... LXF

Скорая помощь

Когда вы будете читать этот материал, уже, возможно, выйдет MonoDevelop версии 1.0, хотя бы в виде бета-версии. Stetic – одна из активно разрабатываемых и быстро меняющихся областей, поэтому в новом релизе будут исправлены многие ошибки и даже добавлены новые... возможности. Стоит проверить!

Chomp 0.2

Версия *Chomp* на диске базируется на том, что мы сделали, и позволяет пользователям временно добавлять новые ленты, но вы можете сделать еще кое-что:

» Заставить работать функцию удаления ленты.
» Кэшировать ленты и GUID, чтобы при отображении они загружались из памяти, а не скачивались каждый раз из Сети.
» Реализовать поиск в кэшированных лентах.
» Добавить Refresh и RefreshAll.
» Поместить что-нибудь в строку состояния!
» Заполнить меню осмысленными действиями.
» Добавлять пользовательские ленты в файл **sitelist.txt**.

Целью этого урока было научить вас использовать *Stetic*, и я думаю, мы справились неплохо. Дальнейшая работа будет в основном писаниной кода: обвешаем интерфейс всякими крутыми штуками. Почаще сохраняйте результат, и все будет в порядке – резвитесь!

» **Через месяц** Мы готовы нырнуть в объектно-ориентированное программирование.

dh_make: Сборка

Ubuntu набирает все большую популярность – почему бы не помочь ему, собрав пакет с одним из недостающих приложений? **Грэм Моррисон** проведет мастер-класс по дебианизации открытого ПО.

Благодаря появлению Ubuntu, Deb-файл стал самым распространенным форматом двоичного пакета в Linux (если верить Google Trends). Большинство дистрибутивов на базе Debian используют Deb-пакеты, а не другие доморощенные форматы упаковки, и хотя мы на данном уроке будем говорить в основном про Ubuntu, эта информация в равной степени применима к прочим дистрибутивам на базе Debian, или к любому другому дистрибутиву, располагающему соответствующими утилитами установки пакетов.

Просьбы о двоичных пакетах буквально засорили Интернет. Пользователи, не имеющие опыта в сборке программного обеспечения, отчаянно рвутся «наложить лапу» на последние версии полюбившегося программного обеспечения. Создание и обмен собственными пакетами – один из лучших видов вклада в сообщество. Приложение, которое я выбрал в качестве примера, Ale – графический инструмент для комбинирования изображений. Новая версия была выпущена в январе, но так как разработчики Ubuntu намерены выпускать только пакеты обновлений по безопасности, создание Deb-файла последнего релиза будет отличным способом добавить новую функцию без мучений по обновлению дистрибутива.

Процесс создания Deb-файла очень схож с созданием RPM и страдает такими же проблемам – нахождением зависимостей вашего пакета. К счастью, в отличие от RPM, существует хороший инструмент, облегчающий создание собственного пакета.

Наш эксперт

Грэм Моррисон
Разработчик свободного ПО, горячий поборник KDE и электронной музыки. Он обсуждал синтезаторы с лидерами сообщества FOSS.

Часть 1: Готовим систему

Большая часть приложений, которую вы скачиваете, использует тройку команд *./configure*, *make* и *make install*, а также систему *Autoconf* для собственной сборки и установки. В своей основе эти инструменты используют набор скриптов для сбора необходимой информации о вашем дистрибутиве, чтобы приложение было правильно скомпилировано, а его файлы попали в нужные каталоги.

Примерно такой же процесс и лежит в создании Deb-файла, а используемая вами система Linux для создания пакетов должна быть возможно чище, чтобы собственные или неподдерживаемые пакеты случайно не попали в число зависимостей. Когда вы создаете пакет из исходных текстов, результирующий двоичный файл будет скомпонован с библиотеками вашей системы. Процесс сборки не знает, оригинальные ли это версии библиотек, собранные вами или скачанные из стороннего репозитория, поэтому необходимо собирать Deb-файлы в системе, максимально близкой к эталонной – и виртуальные машины являются идеальной средой для компиляции программ.

Процесс создания двоичных пакетов аналогичен сборке приложения из исходных текстов, поэтому, прежде чем начать, вам понадобится рабочая среда для компиляции – включая стандартные утилиты сборки GNU – а также библиотеки для разработки, требуемые для сборки и установки выбранного приложения. Вот как вы должны удостоверить-ся, что ваша система содержит все необходимые зависимости для двоичных файлов, которые вы хотите распространять. Установка Ubuntu по умолчанию не содержит пакетов для разработки, но их можно установить, набрав в консоли следующее:

```
sudo APT-get install build-essential
```

Понадобится также установить различные пакеты для разработки, необходимые для корректной компиляции вашего приложения.

Теперь мы добрались до группы дополнительных пакетов, которые используются для создания Deb-файла. Первый – *fakeroot*, он нужен, чтобы файлы в нашем пакете имели привилегии `root`. *Fakeroot* позволяет обычному, не суперпользователю создавать файлы, владельцем которых является `root` – при создании пакетов так безопаснее.

Но львиная доля работы осуществляется пакетом *dh-make*, инструментом, специально спроектированным для «дебианизации» архива исходного кода: он принимает кучу файлов конфигурации и выдает двоичный пакет. Еще два пакета, *debhelper* и *devscripts*, содержат много полезных инструментов. Наконец, последнее, что вам понадобится, это *Dpkg* – в дистрибутивах на базе Debian он установлен по умолчанию, в других же дистрибутивах вам придется самим о нем позаботиться.

Месяц назад Мы пошалили с Grub, добавив картинки, пароли и прочее.

Deb-пакетов

Часть 2: Создаем файл настройки

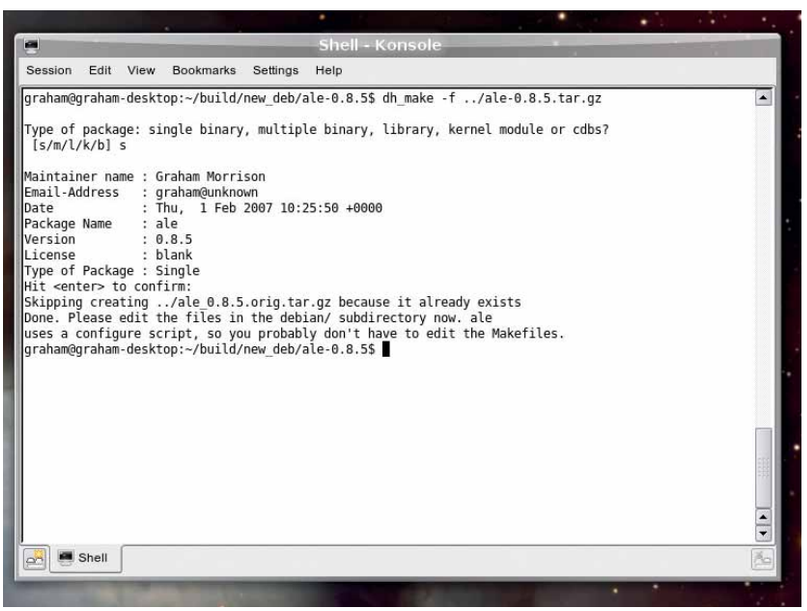
Установив все необходимое для создания Deb-файла, можете скачивать исходный код выбранного приложения. Сперва лучше скомпилировать и запустить приложение, чтобы удостовериться, что ничто не повлияет на процесс сборки. Если все пройдет нормально, возвращайтесь к исходному тексту. Следующий шаг состоит в использовании команды *dh_make*, являющейся частью ранее установленного пакета *dh-make* (дефис, а не знак подчеркивания).

Команда *dh_make* создаст в каталоге с исходным текстом программы подкаталог, в котором будут содержаться все файлы настройки, которые понадобятся *Dpkg* для создания Deb-файла. Чтобы выполнить работу, *dh_make* нужен как каталог, так и архив с исходными текстами в определенном месте с именем определенного формата. Вам потребуется извлечь архив в каталог с названием имя пакета-номер версии – где имя пакета отделено от номера версии дефисом. Для Ale 0.8.5, например, каталог будет назван **ale-0.8.5**. Так уж вышло, что именно этот формат используется большинством приложений (Deb'ами или другими), включая *Ale*, поэтому есть шанс, что каталог будет иметь верное имя по умолчанию.

Переименовав при необходимости каталог с исходным кодом, проверьте, что архив назван согласно тому же формату (т.е. **ale-0.8.5.tar.gz**), и перейдите в каталог с исходным кодом. Для создания необходимых файлов настройки наберите следующую команду и проверьте, что параметр -f указывает на архив с исходными текстами:

```
dh_make -f ../ale-0.8.5.tar.gz
```

Вам потребуется ответить на несколько вопросов. Первый – какой тип пакета вы хотите создать: одиночный двоичный, множественный двоичный, библиотеку, модуль ядра или CDBS. Кроме CDBS, все остальное должно быть понятно. CDBS – это Common Debian Build System [Общая система сборки Debian]; в этом случае создастся основной набор файлов настройки, который потребует значительного редактирования, прежде чем вы сможете создать Deb-файл. Все остальные опции пытаются заполнить файлы настройки автоматически, в зависимости от типа приложения или выбранного инструмента. Для большинства приложений, скачанных из Сети, вы можете спокойно выбрать опцию «одиночный двоичный» (Single Binary). Когда вы сделаете выбор, *dh_make* выведет набор параметров, добытых им из вашей системы, и попросит вас их подтвердить – впрочем, чисто из любезности: пути назад уже нет, и все, что вы можете сделать, это нажать Enter; после чего *dh_make* создаст все файлы настройки и поместит их в каталог **debian**.



› Перед генерацией файлов настройки скрипт *dh_make* задаст вам пару вопросов.

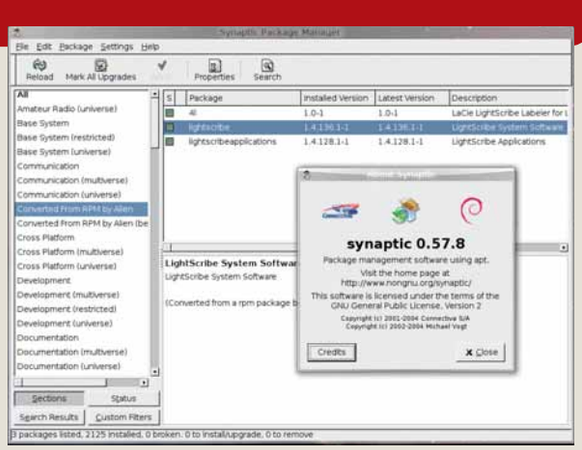
Расставляем все по местам

В прошлом дистрибутив Ubuntu критиковали за использование огромной библиотеки пакетов Debian без обеспечения обратной совместимости со своими пакетами. Но вы можете помочь восстановить баланс сил, собрав собственные пакеты, которые будут работать на любом дистрибутиве.

Создание собственного Deb-пакета требует кое-каких знаний. Но за год-другой работы с Linux вы их накопите. Процесс сборки пакетов – идеальное времяпрепровождение, если вы не умеете программировать, но не боитесь компилировать свое собственное ПО. Пользователи всех уровней часто взывают о последних релизах любимых программ в удобной «упаковке».

Потратив немного времени на создание пакета, вы сбережете время других людей. Только представьте экономию электричества, если тысячи людей мигом установят ваш готовый пакет и не будут собирать его сами!

› Почему бы не поучаствовать в фестивале Ubunty Debfest?



Часть 3: Тонкая настройка

Взглянув на содержимое созданного каталога **debian**, вы увидите бездну настроечных файлов. К счастью, для большинства приложений требуется отредактировать только парочку из них. Первый файл, в который следует заглянуть, это **Control**. Он содержит основную информацию о вашем приложении, а также набор зависимостей для его сборки из исходных текстов (файлы Dev) и установки как бинарного пакета. Многие поля обязательны для заполнения, и *dh_make* должен был уже поработать над ними. Это также относится и к другим необходимым файлам настройки. Очень важен формат этого и всех остальных файлов в каталоге **debian**. Каждое поле располагается на одной строке – никаких возвратов каретки – и между полями Standards-Version и Package должна быть пустая строка. Вот поля, которые придется редактировать:

» Source Имя вашего приложения. В нашем примере это *Ale*.

» Section Категория вашего приложения. Нажмите на кнопку Разделы в *Synaptic*, чтобы посмотреть примеры.

» Maintainer Так как вы поддерживаете данный пакет, здесь следует указать ваше имя и адрес электронной почты.

» Standards Это поле следует оставить как есть, так как оно представляет версию стандарта упаковки Debian, используемую для файлов настройки.

» Package Укажите здесь то же, что и в поле Source.

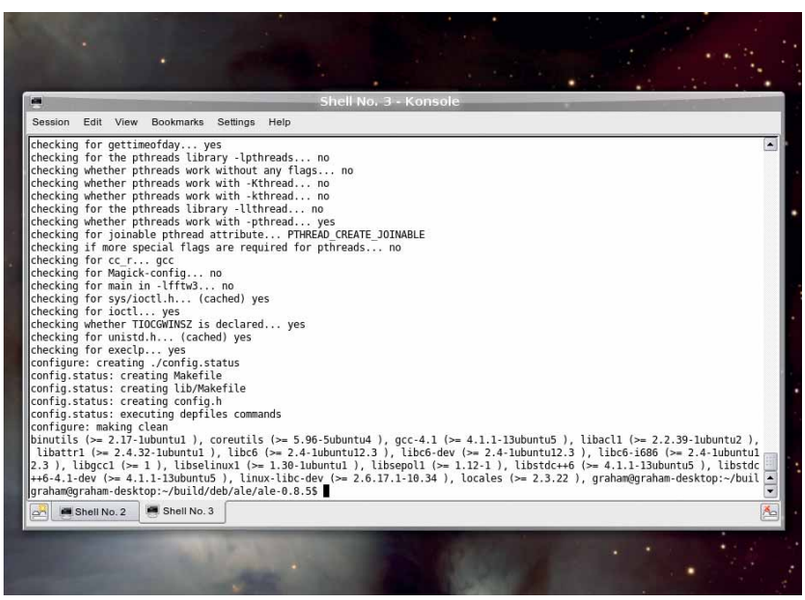
» Architecture Если вы хотите создать пакет для конкретной архитектуры процессора, то указать ее следует здесь. Это может быть, к примеру, i386 или AMD64, но для сборки пакета вам потребуется именно этот процессор. Вы можете проверить архитектуры, поддерживаемые вашим процессором, набрав **dpkg-architecture** в консоли.

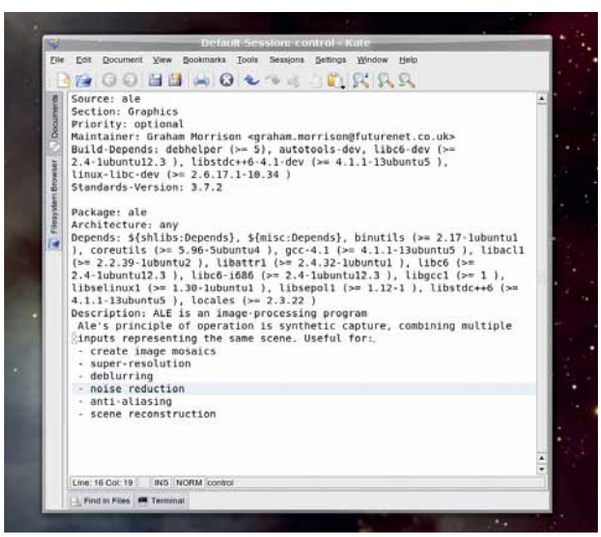
» Description В этом поле формат особенно некстати: требуется описать ваш пакет менее чем 60-ю символами. Но можно продолжить описание, только последующие строки должны начинаться с пробела, а уж за ним помещайте столько текста, сколько необходимо.

Если вы, читая это руководство, одновременно смотрите на файл настройки, то наверняка увидели, что я пропустил два важных поля: Build-Depends и Depends. Они описывают зависимости: Build-Depends – список всего, что необходимо для компиляции и сборки пакета из исходных текстов, а Depends – список других двоичных пакетов, требующихся для правильной работы вашего приложения. Есть много способов, позволяющих определить нужные зависимости – включая просмотр документации к приложению, скриптов настройки или просто зависимостей старой версии в менеджере пакетов *Synaptic*; но разработчики Debian придумали кое-что получше: скрипт, который, получив имя вашего приложения, выдаст список всех его зависимостей. Скрипт полагается на то, что собираемое вами приложение является корректным проектом *Autotools*:

```
strace -f -o /tmp/log ./configure
# или же make вместо ./configure, если пакет не использует autoconf
```

› Можете использовать скрипт для поиска зависимостей вашего приложения – они будут показаны в конце вывода.





› Это наш файл настройки политики – в нем все зависимости, добавленные для сборки приложения.

Альтернатива: CheckInstall

Есть более легкий способ создать Deb-файлы: использовать *CheckInstall*. Эта отличная утилита требует меньше трудов, чем *dh_make*, потому что не надо редактировать никаких файлов настройки. Но за счет чего? Она не такая гибкая, и будет работать только с более общим классом приложений, которые используют *Autoconf*.

Установив *CheckInstall* с помощью менеджера пакетов вашего дистрибутива, скачайте и распакуйте исходный код приложения – кандидата на «пакетирование». На самом деле, *CheckInstall* подключается на стадии *make install* сборки вашего приложения, перехватывая контроль над процессом для получения списка зависимостей от библиотек, создаваемых во время компиляции. Чтобы добраться до этой стадии, наберите *./configure*, затем *make* в каталоге с исходным текстом вашего приложения. А вместо *make install* наберите *checkinstall*.

Вам зададут пару вопросов (они служат той же цели, что и информация, содержащаяся в файле **Control**), включая путь к документации и описанию вашего пакета, и вы получите шанс отредактировать глобальные параметры, например, описание пакета и лицензию. После этого *CheckInstall* создаст Deb-файл и поместит его в каталог с исходным текстом.

› Пара простых вопросов, и CheckInstall создаст Deb-файл автоматически.

```
for x in `dpkg -S $(grep open /tmp/log|\
perl -pe 's!.* open\("([^\"]*).*!$1!' |\
grep "^/"|sort | uniq|\
grep -v "^\(/tmp\|/dev\|/proc\)" ) 2>/dev/null|\
cut -f1 -d":"|sort | uniq`; \
do \
echo -n "$x (>=" `dpkg -s $x|grep ^Version|cut -f2 -d":"` "), "; \
done
```

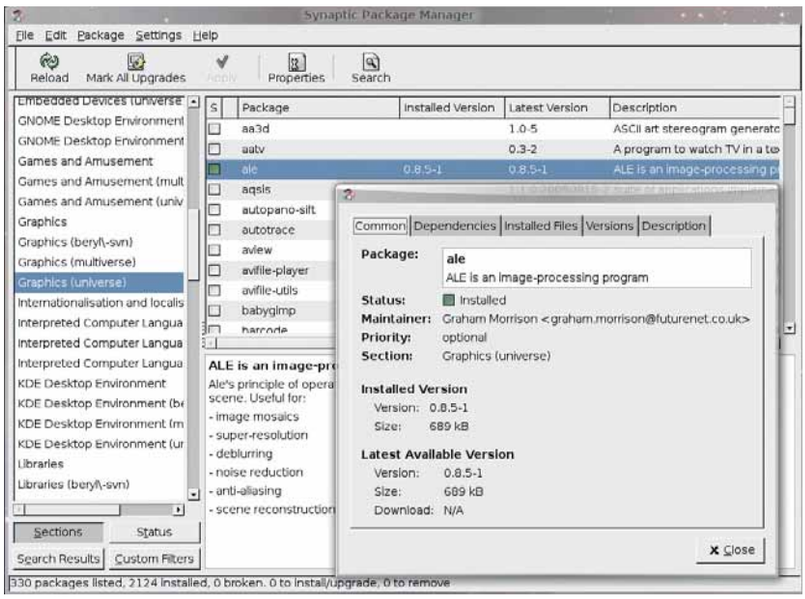
Сохраните этот код в текстовом файле и запустите его в каталоге, содержащем файл настройки вашего приложения (просто наберите **sh ./имя_файла**). Его вывод подобен создаваемому скриптом **./configure**, но отфильтрован с тем, чтобы получить оригинальные *Dpkg*-имена зависимостей. Они появятся в самом конце вывода, после строки 'configure:make clean', и их надо скопировать в поля Build-Depends и Depends файла **Control**. Имена пакетов для разработки оканчиваются на '-dev', они добавляются в строку Build-Depends, включая строку с версией, следующую за именем пакета – **libstdc++6-4.1-dev(>=4.1.1-13ubuntu5)**, например. Все остальные зависимости надо добавить в поле Depends.

Если ваше приложение использует троицу *Autotools* из команд *configure*, *make* и *install* для вашего пакета, вам не нужно предоставлять еще какую-то информацию. Без *Automake* понадобится редактировать файл **rules**, а это само по себе может потребовать руководства на четыре страницы. Фактически это Debian-версия файла **Makefile**, которую **dpkg-buildpackage** использует для сборки вашего пакета. Но в остальном трудности позади. Теперь надо только отредактировать несколько других настроечных файлов, а потом собрать бинарный пакет.

Откройте файл **changelog** в текстовом редакторе. Этот файл – Debian-эквивалент рабочего журнала программиста: места, куда помещаются отличия данного релиза от предыдущего. Изменения заносятся в раздел, следующий за символом звездочки, и вам также надо включить сюда ваше имя и адрес электронной почты. Основная цель этого файла – дать вашему пакету имя, распознаваемое менеджером пакетов Debian. Текущее имя находится в первой строке файла настройки, и

> «Небольшой скрипт выдаст вам зависимости вашего приложения.»

его надо отредактировать в соответствии с соглашениями по именованию, принятыми в вашем дистрибутиве. Для Ubuntu единственной заменой будет замена слова 'unstable' на 'edgy', точнее, на ту версию дистрибутива, которую вы используете. Новое Руководство Debian для ответственных за программы (www.debian.org/doc/maint-guide) также предлагает отредактировать файлы **Copyright** и **README.debian**, включив всевозможную информацию о пакете, который вы собираете. Скрипт *dh_make* создал шаблоны man-страниц вашего приложения, и их надо перезаписать реальными файлами, идущими в его составе, или же создать руководство самому, если вы думаете, что у вас получится (*см.* LXF76 – там рассказано, как писать man-страницы). Есть еще файл инициализации системы (**init.d.ex**), он может использоваться для запуска сервисов во время загрузки, если ваше приложение того требует, а также довольно избыточные шаблоны *Emacs* (**emacsen***) и набор скриптов (**pre*** и **post***). Если вы достаточно хорошо знаете собираемое приложение, то сумеете извлечь выгоду из дополнительных возможностей. В противном случае удалите файлы, которые не хотите использовать, чтобы они не установились по умолчанию.



› Установив Deb-файл, вы можете удалить его с помощью менеджера пакетов вашего дистрибутива.

Часть 4: Запуск системы

Файлы настройки отредактированы, и все готово к финальному аккорду – сборке Deb-файла. Выполните магическую команду **dpkg_buildpackage -rfakeroot** из каталога с исходным кодом. Запустившись, она скомпилирует программу и свяжет вместе различные компоненты, используя файлы настройки. По окончании ее работы вы найдете в родительском каталоге с исходным кодом группу файлов, включая главный двоичный Deb-пакет. Это и будут те самые файлы, которые можно установить и передавать другим людям. Для установки нового пакета просто наберите **sudo dpkg -i ale_0.8.5-1_i386.deb**. Потом вы сможете использовать ваше приложение, а если оно вам надоест, удалить его с помощью менеджера пакетов вашего дистрибутива.

Теперь у вас есть Debian-эквивалент информационной бомбы на вашем жестком диске, осталось только швырнуть ее в других пользователей (сосчитав до трех). Хорошая стартовая площадка – форум вашего дистрибутива. Вы и охнуть не успеете, как легионы преданных пользователей примутся скачивать, устанавливать и использовать ваш пакет – забивая ваш почтовый ящик вопросами, дальнейшими пожеланиями, жалобами, а то и шальными «спасибо».

Добро пожаловать в сообщество Open Source! LXF

Скорая помощь

Если ваше приложение содержится только в RPM-пакете, а Deb-файлов не предоставляет, собирать его вручную не обязательно: воспользуйтесь инструментом под названием *Alien*. Просто наберите **alien --to-deb** пакет.rpm и установите получившийся Deb-файл с помощью **dpkg -i пакет.deb**.

» **Через месяц** Мы разберемся с Java: Ant, Javadoc и SFTP.

Glade на новый

ЧАСТЬ 5 Описать интерфейс программы в XML и подгрузить его на лету? Совершенно необязательно покупать Visual Studio и устанавливать .NET 3.0 – **Андрей Боровский** покажет, как решить задачу средствами Glade!

Компьютеры бесполезны, ибо они дают только ответы.

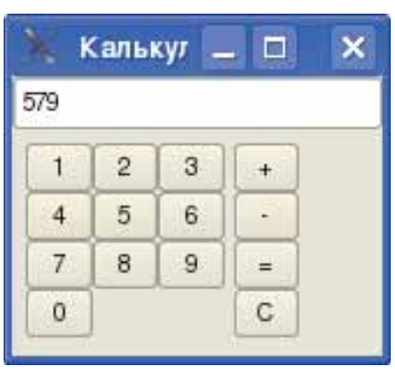
Сальвадор Дали

Если бы великий художник лучше разбирался в компьютерах, он сумел бы найти в них гораздо больше недостатков. Чем занимаются большую часть времени настольные компьютеры? Они выполняют работу более примитивных устройств. Компьютер может быть видеомагнитофоном, радиоприемником, будильником и, конечно, конечно, пишущей машинкой. Ну а кроме всего прочего, любая операционная система на любой платформе предлагает программу, выполняющую функции простейшего микрокалькулятора. Непреложный факт – микрокалькулятор обязательно будет сэмулирован на любой системе, достаточно мощной доля того, чтобы его эмулировать. Нет, я, конечно, понимаю, что компьютерные проигрыватели, компьютерные будильники и компьютерные микрокалькуляторы полезны. Я и сам с удовольствием ими пользуюсь. Я лишь хочу сказать, что компьютеры все еще по-настоящему не революционизировали нашу жизнь. Ну а в ожидании революции, после которой все будет не так, как было, мы, уважаемый читатель, тоже напишем простейшую программу-калькулятор, и воспользуемся для этого, естественно, инструментарием *GTK+*.

В предыдущей статье мы научились создавать проекты графических приложений с помощью *Glade*. Используя *Glade 2.x*, мы сгенерировали не только описание интерфейса программы, но и заготовку ее исходного кода, включая вспомогательные функции и обработчики событий. Сегодня мы рассмотрим другой способ работы с *Glade*. При новом подходе *Glade* используется исключительно для проектирования интерфейса, а исходный код приложения пишется программистом с помощью других средств (по-видимому, этот способ работы станет единственно возможным в *Glade 3.x*).

Описание интерфейса, созданное *Glade* в формате XML, и исходный код программы связываются между собой с помощью функций библиотеки *libglade*. Все это немного похоже на систему XMLGUI, реализованную в KDE. Разница заключается в том, что в XMLGUI внешний XML-документ описывает лишь отдельные элементы интерфейса программы, тогда как при работе с *libglade* все описание интерфейса загружается из XML-файла [Это не означает, что в Qt/KDE нет технологии, аналогичной libglade. Интерфейсы, созданные в Qt Designer, также можно загружать и обрабатывать на лету, – прим. ред.].

(Рис. 1) Программа-микрокалькулятор.



Приступим к проектированию

Для того, чтобы понять, как работает библиотека *libglade*, необходимо сначала разобраться, что представляет собой проект Glade. Разбираться, как вы уже поняли, мы будем на примере программы-микрокалькулятора (**Рис. 1**).

Как можно заметить, наш калькулятор довольно примитивен. Поддерживаются только операции сложения и вычитания, причем над целыми числами. Отсутствует контроль переполнения при выполнении арифметических действий. При желании вы можете нарастить функциональность нашего микрокалькулятора сами – добавить поддержку чисел с плавающей точкой, тригонометрические функции, встроенный язык программирования... Нет пределов совершенству. Мы же сосредоточимся на программировании интерфейса калькулятора средствами *GTK+*, *Glade* и *libglade*.

Как это обычно бывает при работе со средствами визуального программирования, проектирование программы начинается с пользовательского интерфейса. В главном окне приложения мы размещаем несколько контейнеров для более удобной упаковки элементов калькулятора – индикаторной панели и кнопок.

Свойству «Имя» объекта главного окна мы присваиваем значение rootwnd. Это имя нам нужно запомнить, так как оно будет играть важную роль при взаимодействии с библиотекой *libglade*. В верхней части главного окна расположено текстовое поле GtkEntry, которое будет служить индикаторной панелью калькулятора. Кнопки калькулятора – это объекты GtkButton. Вот, собственно, и все, других элементов управления наш виртуальный калькулятор (как и его «железный» собрат) не предполагает.

Иерархию объектов-контейнеров и элементов управления, из которых состоит калькулятор, проще (и полезнее) показать, нежели описать. Для этого нужно всего лишь открыть окно «Дерево эл. управления» среды *Glade* (**Рис. 2**). Перед вами появится древовидный список всех визуальных элементов.

Объекты-контейнеры являются внутренними узлами дерева, а элементы управления – его «листьями». Если мы теперь сохраним проект *Glade* под именем calculator, на диске появится файл **calculator.glade**. Именно этот файл содержит описание спроектированного нами графического интерфейса программы. Если мы откроем файл **calculator.glade** в текстовом редакторе, то увидим примерно следующее:

```
<?xml version="1.0" standalone="no"?> <!--*- mode: xml -*-->
<!DOCTYPE glade-interface SYSTEM "http://glade.gnome.org/glade-2.0.dtd">

<glade-interface>

<widget class="GtkWindow" id="rootwnd">
  <property name="visible">True</property>
  <property name="title" translatable="yes">Калькулятор</property>
```

Месяц назад Мы изучали визуальное программирование в духе старой школы *Glade 2.x*.

ЛАД

```
...
  <signal name="destroy" handler="gtk_main_quit" last_modification_
 time="Tue, 13 Mar 2007 21:18:21 GMT"/>

  <child>
    <widget class="GtkVBox" id="vbox1">
      <property name="visible">True</property>
      ...
      <child>
              <widget class="GtkEntry" id="entry1">
                <property name="visible">True</property>
                ...
      </child>
...
```

Таким образом, файл ***.glade** представляет собой XML-документ, в котором содержатся списки значений свойств каждого объекта интерфейса, а иерархия интерфейса реализована при помощи иерархии вложенных тегов.

Немного кода

Теперь, когда у нас есть описание графического интерфейса, созданное *Glade*, нам будет очень просто написать демонстрирующую его программу. Открою небольшой секрет – функциональность *libglade* используется самой средой *Glade*, которая должна отображать создаваемый интерфейс в отдельном окне. Исходный код программы, загружающей и отображающей графический интерфейс, описанный в файле calculator.glade, состоит примерно из десяти строк (вы найдете его на диске в файле **calculator-1.c**).

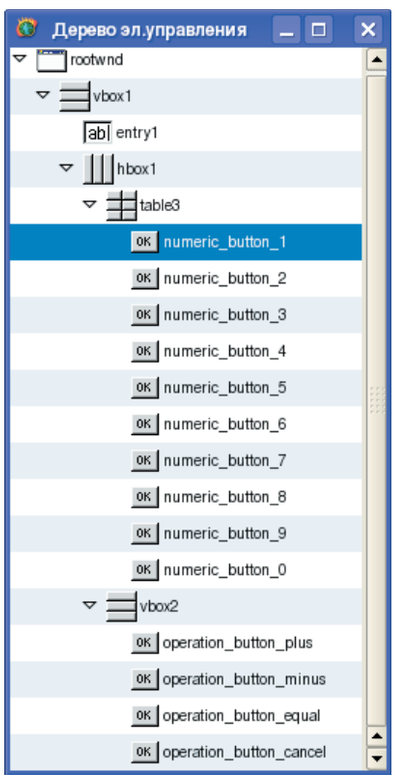
```
#include <stdlib.h>
#include <stdio.h>
#include <gtk/gtk.h>
#include <glade/glade.h>

int main (int argc, char **argv)
{
  GladeXML *xml;
   gtk_init(&argc, &argv);
   glade_init();
   xml = glade_xml_new("calculator.glade", "rootwnd", NULL);
   if (!xml) {
            g_warning("Failed to create the interface");
            return 1;
   }
   gtk_main();
   return 0;
}
```

Обратите внимание, что теперь, помимо заголовочного файла **gtk/gtk.h** мы включаем в программу файл **glade/glade.h**. Он содержит объявления функций и типов данных библиотеки *libglade*, а именно она будет служить рабочей лошадкой нашей программы. Собственно программа начинается с вызова знакомой нам функции gtk_init(). Далее мы вызываем новую функцию glade_init(). Ее задача заключается в том, чтобы инициализировать систему *libglade*. Связывание программы с XML-файлом, описывающим ее интерфейс, выполняется функцией glade_xml_new(). Первым аргументом этой функции должен быть файл, созданный в *Glade*, вторым – имя корневого элемента иерархии графических объектов интерфейса, определенного в этом файле. Третий параметр glade_xml_new() описывает домен перевода приложения (translation domain). В качестве значения этого параметра мы можем передать NULL. Функция glade_xml_new() возвращает указатель на структуру *GladeXML*, который мы сохраняем в переменной xml. Структура *GladeXML* инкапсулирует описание интерфейса, созданного *Glade*. Библиотека *libglade* экспортирует несколько функций, имена которых начинаются с префикса glade_xml_. Эти функции позволяют управлять элементами интерфейса, созданного с помощью *Glade*, и каждой из них в качестве одного из параметров следует передавать указатель на структуру *GladeXML*.

Впрочем, в версии **calculator-1.c** значение переменной **xml** еще не востребовано. Вызов glade_xml_new() приводит к загрузке файла описания интерфейса в программу, а библиотека *libglade* позаботится о его правильном отображении. Нам остается только запустить цикл обработки сообщений с помощью функции gtk_main(). Скомпилируем нашу программу, используя следующую команду:

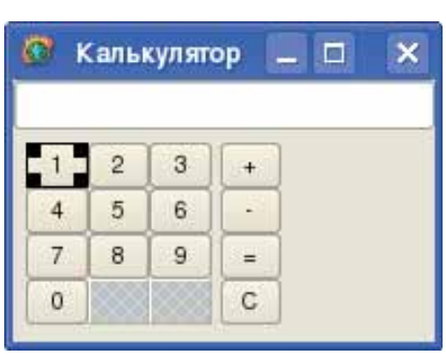
```
gcc calculator-1.c -o calculator `pkg-config --cflags --libs libglade-2.0`
```

На этот раз в качестве аргумента утилиты *pkg-config* мы используем имя пакета *libglade-2.0*. Теперь можно запустить программу *calculator*. Вы, конечно, сразу заметите, что хотя программа правильно отображает все элементы пользовательского интерфейса, она ничего не делает (даже завершиться как следует не умеет). Это вполне естественно, ведь мы еще не определили ни одного обработчика сигнала.



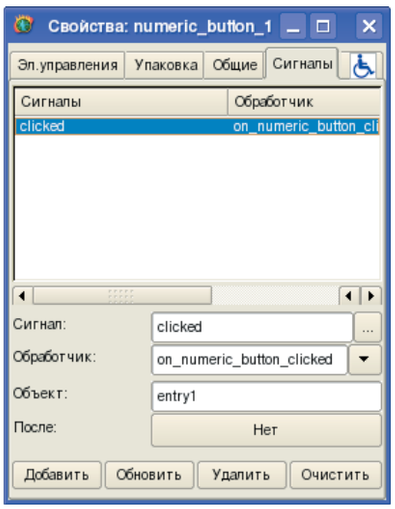
(Рис. 2) Окно просмотра структуры элементов интерфейса.

Снова откройте (если, вдруг, вы его закрыли) проект *calculator* в среде *Glade*. Прежде всего, выделим в редакторе свойств объект rootwnd, перейдем на вкладку «Сигналы» редактора свойств и создадим обработчик сигнала destroy. В качестве процедуры обработчика из раскрывающегося списка «Обработчик» выберем функцию gtk_main_quit(). Вот так просто можно связать обработчик сигнала закрытия окна и функцию завершения работы программы. Далее выделим в редакторе форм одну из цифровых кнопок калькулятора (**Рис. 3**).



(Рис. 3) Окно редактора формы с выделенной кнопкой.

Всем цифровым кнопкам присвоены имена вида numeric_button_x,



(Рис. 4) Окно редактора свойств в режиме создания обработчика сигнала.

где x – число от 0 до 9. Перейдем на вкладку «Сигналы» редактора свойств и создадим для выбранной кнопки обработчик сигнала clicked. Процедура обработчика получит имя on_numeric_button_x_clicked(). Переименуем обработчик в on_numeric_button_clicked(). При назначении обработчика каждой кнопке в поле ввода «Объект» укажем значение entry1 (**Рис. 4**).

Напомню, что поле ввода «Объект» позволяет задать имя объекта, указатель на который будет передан процедуре обработчика в качестве дополнительного параметра. Объект entry1 – это объект класса GtkEntry – «индикаторная панель» нашего калькулятора. Вполне естественно, что обработчики событий clicked цифровых кнопок должны иметь доступ к объекту, представляющему индикаторную панель. Мы переименовали обработчик сигнала clicked в on_numeric_button_clicked(), чтобы подчеркнуть, что у нас будет один обработчик сигнала clicked для всех цифровых кнопок.

Назначим теперь обработчик on_numeric_button_clicked() сигналам clicked всех кнопок с цифрами (не забудем указать объект entry1 в качестве дополнительного параметра обработчика каждой кнопки). Кнопки +, –, = и C имеют имена operation_button_plus, operation_button_minus, operation_button_equal и operation_button_cancel соответственно. В среде *Glade* создадим для каждой кнопки свой обработчик события clicked (соответственно функции on_operation_button_plus_clicked(), on_operation_button_minus_clicked(), on_operation_button_equal_clicked() и on_operation_button_cancel_clicked()). Обработчикам on_operation_button_equal_clicked() и on_operation_button_cancel_clicked() в качестве дополнительного параметра должен быть передан объект entry1. Сохраним проект *Glade*. На этом визуальное программирование калькулятора закончено, и нам остается довершить воплощение наших идей в коде. Вариант программы с обработчиками сигналов вы найдете в файле **calculator-2.c**, основанном на файле **calculator-1.c**.

Доведем до совершенства!

Прежде всего нам нужно написать функции-обработчики сигналов:

```
void on_numeric_button_clicked(gpointer user_data, GtkButton *button)
{
  int i = atoi(gtk_button_get_label(button));
  if (mode == COPY_MODE) {
    bgvalue = fgvalue;
    fgvalue = 0;
    mode = INPUT_MODE;
  }
  fgvalue = fgvalue*10 + i;
  sprintf(screen, "%li\0", fgvalue);
  gtk_entry_set_text(GTK_ENTRY(user_data), screen);
}

void on_operation_button_plus_clicked(GtkButton *button)
{
  mode = COPY_MODE;
  op_state = OP_PLUS;
}

void on_operation_button_minus_clicked(GtkButton *button)
{
  mode = COPY_MODE;
  op_state = OP_MINUS;
}

void on_operation_button_equal_clicked(gpointer user_data, GtkButton
*button)
{
  switch (op_state) {
    case OP_PLUS:
      fgvalue += bgvalue;
      break;
    case OP_MINUS:
      fgvalue = bgvalue - fgvalue;
    default:;
  }
  sprintf(screen, "%li\0", fgvalue);
  gtk_entry_set_text(GTK_ENTRY(user_data), screen);
  mode = COPY_MODE;
}

void on_operation_button_cancel_clicked(gpointer user_data, GtkButton
*button)
{
  bgvalue = fgvalue = 0;
  mode = COPY_MODE;
  op_state = OP_NONE;
  sprintf(screen, "%li\0", fgvalue);
  gtk_entry_set_text(GTK_ENTRY(user_data), screen);
}
```

Необходимо указать одну особенность взаимодействия программы *libglade* с обработчиками сигналов. Если при связывании сигнала с обработчиком указывается дополнительный параметр, этот параметр будет первым аргументом функции-обработчика. Вторым аргументом будет указатель на объект-источник сигнала. Например, в обработчике on_numeric_button_clicked() первый аргумент указывает на объект entry1, а второй аргумент – на объект numeric_button_x, для которого вызван обработчик. Если же при связывании обработчика с сигналом дополнительные параметры не указываются, первым (и единственным) аргументом функции-обработчика будет указатель на объект-источник сигнала. Например, в обработчике on_operation_button_plus_clicked() первым аргументом является указатель на объект operation_button_plus, для которого вызван обработчик.

Вдаваться в подробности работы калькулятора мы не будем. Во-первых, они очевидны. Во-вторых, наша статья посвящена не написанию калькуляторов, а использованию *libglade*, так что мы рассмотрим лишь некоторые аспекты взаимодействия обработчиков с элементами интерфейса.

В обработчике on_numeric_button_clicked() нам необходимо получить цифру, соответствующую нажатой кнопке. Мы не можем передавать цифру в дополнительном параметре обработчика, поскольку этот параметр уже занят объектом entry1. Вместо этого мы просто считываем цифру, являющуюся меткой нажатой кнопки, с помощью функции gtk_button_get_label(). Текст элемента ввода entry1 устанавливается с помощью функции gtk_entry_set_text().

Хотя мы и создали обработчики всех сигналов, но если бы мы скомпилировали программу на данном этапе, ее элементы управления все равно ничего бы не делали. Для того чтобы наладить в программе обработку сигналов, недостаточно написать их обработчики. Необходимо решить еще одну проблему, которая в явном или неявном виде возникает во всех средах визуального программирования. В описании интерфейса программы мы указали, например, что обработчиком события clicked для кнопки operation_button_minus является нечто по имени on_operation_button_minus_clicked(). На этапе визуального программирования строка on_operation_button_minus_clicked() не является именем функции, поскольку никакой функции еще нет. Фактически мы просто добавили в описание интерфейса программы имя обработчика сигнала. Если бы связывание сигнала и обработчика выполнялось на этапе компоновки программы (как это происходит в «настоящих» IDE), компоновщик нашел бы функцию on_operation_button_minus_clicked() и связал бы ее с сигналом. Но в *GTK+* связывание сигналов и обработчиков происходит во время выполнения программы. Каким же образом программа находит функцию, соответствующую имени обработчика?

Ведь на этапе выполнения программы система *libglade* не знает имен функций (а знает только их адреса) и не может сопоставить функцию с именем обработчика. Для решения этой проблемы мы должны явным образом указать, что функция on_operation_button_minus_clicked(), определенная нами в файле **calculator-2.c**, является «тем самым» обработчиком on_operation_button_minus_clicked(). Мы делаем это с помощью функции glade_xml_signal_connect(). Первым параметром glade_xml_signal_connect() должен быть указатель на структуру *GladeXML*. Вторым параметром функции является имя обработчика, заданное при описании интерфейса в среде *Glade*, а третьим параметром – адрес определенной нами функции-обработчика. Из сказанного следует, между прочим, что функция-обработчик сигнала не обязана иметь то же имя, что и обработчик, назначенный сигналу в файле описания интерфейса. Важно лишь установить соответствие между двумя именами (точнее, между именем обработчика и адресом функции). Таким образом, в функцию main() файла **calculator-2.c** нужно добавить следующие строки, в которых устанавливается соответствие между именами обработчиков, заданными в файле **calculator.glade**, и функциями-обработчиками, определенными в программе:

```
glade_xml_signal_connect(xml, “gtk_main_quit”, G_CALLBACK(gtk_main_quit));
  glade_xml_signal_connect(xml, “on_numeric_button_clicked”, G_CALLBACK(on_numeric_button_clicked));
  glade_xml_signal_connect(xml, “on_operation_button_plus_clicked”, G_CALLBACK(on_operation_button_plus_clicked));
  glade_xml_signal_connect(xml, “on_operation_button_minus_clicked”, G_CALLBACK(on_operation_button_minus_clicked));
  glade_xml_signal_connect(xml, “on_operation_button_equal_clicked”, G_CALLBACK(on_operation_button_equal_clicked));
  glade_xml_signal_connect(xml, “on_operation_button_cancel_clicked”, G_CALLBACK(on_operation_button_cancel_clicked));
```

Теперь вы можете скомпилировать программу и убедиться, что все кнопки (включая кнопку закрытия окна, завершающую приложение) работают как надо.

Помимо функции glade_xml_signal_connect(), библиотека *libglade* экспортирует функцию glade_xml_signal_connect_data(). От glade_xml_signal_connect() эта функция отличается тем, что позволяет указать объект, который будет передаваться обработчику сигнала в качестве дополнительного параметра. В нашей программе мы определили передачу объекта entry1 обработчикам сигналов еще на этапе визуального программирования. Однако может случиться так, что объект, который следует передать обработчику сигнала, станет известен только во время выполнения программы. В этой ситуации функция glade_xml_signal_connect_data() окажется более полезной.

API libglade предоставляет нам еще несколько функций, связанных с обработкой сигналов. Функция glade_xml_signal_autoconnect() сканирует файл ***.glade** в поисках сигналов и функций-обработчиков, автоматически связывая сигналы с назначенными им обработчиками, если обработчики являются функциями API *GTK+* (например, эта функция может связать сигнал destroy и функцию-обработчик gtk_main_quit). Неудобство, связанное с использованием glade_xml_signal_autoconnect() (простите за невольный каламбур) заключается в том, что эта функция будет выдавать предупреждающее сообщение всякий раз, когда ей встретится функция-обработчик, не являющаяся частью API *GTK+*. Определенный интерес представляют также функции glade_xml_signal_connect_full() и glade_xml_signal_autoconnect_full(). Первым параметром каждой из этих функций служит, как обычно, указатель на структуру *GladeXML*. Вторым параметром обеих функций должен быть указатель на определенную программистом функцию *GladeXML*ConnectFunc(), которая будет осуществлять связывание сигналов с обработчиками. Таким образом, связывание сигналов и обработчиков как бы делегируется другой функции, не являющейся частью API Glade XML. Такое делегирование может быть полезно при интеграции API Glade XML с другими языками программирования.

В примере из предыдущей статьи в нашем распоряжении имелась удобная функция lookup_widget(), которая позволяла найти указатель на объект визуального элемента по его имени. Эта функция не является частью API *GTK+*, система генерации кода *Glade 2.x* сама создала ее для нас. Рад сообщить вам, что при работе с библиотекой libglade у нас есть возможность использовать похожую функцию glade_xml_get_widget(). Эта функция возвращает указатель на визуальный объект, заданный своим именем. Первым аргументом glade_xml_get_widget() должен быть указатель на структуру *GladeXML*, вторым аргументом – имя искомого объекта. Конечно, для того, чтобы мы могли воспользоваться функцией glade_xml_get_widget() в обработчике сигнала, переменная-указатель на структуру *GladeXML* должна быть в области видимости процедуры-обработчика. Эту проблему можно решить, объявив переменную-указатель на GladeXML как статическую. Этот метод, однако, неудобен, если обработчик сигнала расположен не в том же файле, что и функция main(). В общем случае указатель на структуру *GladeXML* можно получить другим способом. Если у нас есть указатель на один из визуальных элементов, образующих интерфейс (а в обработчике сигнала как минимум один такой указатель есть всегда), мы можем воспользоваться функцией glade_get_widget_tree(). Аргументом функции glade_get_widget_tree() должен быть указатель на объект, представляющий элемент интерфейса, а ее возвращаемым значением будет указатель на структуру *GladeXML*. Вот как, например, можно получить доступ к этой структуре в некотором обработчике on_button_clicked():

```
void on_button_clicked(GtkButton *button)
{
  GladeXML *xml = glade_get_widget_tree(GTK_WIDGET(button));
...
}
```

Материал, рассмотренный в этой, равно как и в четырех предыдущих статьях, не охватывает всего многообразия программирования *GTK+*. Но, я надеюсь, что эти статьи создали нечто вроде стартовой площадки, которая поможет вам вознестись к настоящим высотам, если программирование *GTK+* вас заинтересует. Мы же перейдем к освоению премудростей программирования для основанной на *GTK+* среды GNOME. LXF

Через месяц До свидания, *GTK+*, здравствуй, GNOME! Что бы там ни говорил Линус...

ncurses:

ЧАСТЬ 11 Оконный интерфейс не обязательно должен быть графическим! **Андрей Боровский** покажет, как создавать удобные приложения, работающие прямо в консоли.

As honour, love, obedience, troops of friends,
I must not look to have; but, in their stead,
Curses, not loud but deep, mouth-honour, breath,
Which the poor heart would fain deny, and dare not.

William Shakespeare, The Tragedy of Macbeth.

В прошлый раз мы научились управлять текстовой консолью с помощью интерфейса *termios*. Однако, для того, чтобы представить текстовый экран во всем великолепии, возможностей *termios* недостаточно. Сегодня мы поговорим о дополнительном средстве управления терминалом – библиотеке *ncurses*. Она и вправду заставляет терминал переливаться всеми цветами радуги (вот почему во всей серии статей из серии Unix API, эта – единственная, в которой вы найдете снимки экранов).

Когда-то давным-давно графические терминалы были редкостью, а пользователи текстовых терминалов хотели работать с интерфейсами, похожими на графические (и, самое главное, использовать новое удобное средство ввода – мышь). Специально для того, чтобы предоставить интерфейс «покажи и щелкни» пользователям текстовых терминалов, была разработана библиотека curses (ее название происходит от ее важнейшей функции – управления курсором, а вовсе не от проклятия, которое она накладывает на программистов). Изначально библиотека *curses* создавалась для BSD UNIX. В Linux используется открытый (распространяющийся на условиях MIT License) клон *curses* – библиотека *ncurses* (new curses). Приложений, использующих *ncurses* в современной Linux-системе не так уж и много. Среди наиболее популярных проектов на базе *ncurses* можно назвать *Midnight Commander*, текстовый Web-браузер *lynx*, программу для чтения новостей *tin* и почтовый клиент *mutt*. Сравнительно невысокая популярность *ncurses* объясняется тем, что ниша ее применения сократилась. Большинство современных компьютеров поддерживают растровую графику, так что если вам нужно реализовать интерфейс «покажи и щелкни», вы, как правило, можете воспользоваться более совершенными графическими средствами. Выбирать *ncurses* как платформу для нового проекта следует, только если, с одной стороны, вашей программе совершенно необходим интерфейс «покажи и щелкни», а с другой – она должна работать в условиях полного отсутствия графической подсистемы.

Введение в ncurses

Основными понятиями в *ncurses* являются экран (screen), окно (window) и подокно (subwindow). Экраном называется все пространство, на котором *ncurses* может выводить данные. С точки зрения *ncurses*, экран – это матрица ячеек, в которые можно выводить символы. Если монитор работает в текстовом режиме, экран *ncurses* совпадает с экраном монитора. Если терминал эмулируется графической программой, экраном является рабочая область окна этой программы. Окном *ncurses* называется прямоугольная часть экрана, для которой определены особые параметры вывода. В частности, размеры окна влияют на перенос и прокрутку выводимых строк. В каком-то смысле окно можно назвать «экраном в экране». В процессе инициализации *ncurses* автоматически создается окно stdscr, размеры которого совпадают с размерами экрана. Кроме структуры stdscr по умолчанию создается еще одна структура – curscr. Если окно *stdscr* предназначено в *ncurses* для стандартного вывода данных, то curscr содержит копию данных, отражающую текущее состояние экрана. Кода вы записываете данные в stdscr (или другое окно), эти данные не отображаются на экране монитора автоматически. Для того, чтобы сделать новый вывод видимым, вы должны вызывать специальную функцию обновления экрана (refresh() или wrefresh()). Эта функция сравнивает содержимое окна *stdscr* и curscr и обновляет экран на основе различий между ними, а затем вносит изменения в структуру curscr. Благодаря наличию окна curscr, приложению *ncurses* не требуется «помнить» весь свой предыдущий вывод и перерисовывать его всякий раз, когда в этом возникает потребность. Этим программы *ncurses* отличаются от графических программ. В старину, когда терминалы связывались с компьютерами через модемы, использование двух окон давало дополнительное преимущество в скорости обмена данными, ведь программе нужно было передавать на терминал не копию экрана целиком, а только последние изменения.

Помимо стандартных окон *ncurses*, вы можете создавать собственные окна размера, меньшего stdscr. Ваша программа может работать с несколькими окнами одновременно, выполняя вывод в каждое из них. Кроме окон (windows), программы *ncurses* могут создавать подокна (subwindows), поведение которых несколько отличается от поведения стандартных окон.

Важнейшей особенностью *ncurses* является возможность указать произвольную позицию курсора для вывода (и ввода) данных. Позиция кур-

Месяц назад Мы управляли терминалом и изучали консольный ввод-вывод.

привет Окнам!

сора отсчитывается от левого верхнего угла текущего окна. Ячейка в верхнем левом углу имеет координаты (0, 0). При работе с функциями *ncurses* важно помнить, что первой координатой является номер строки (что соответствует y в терминах графического программирования), а второй координатой – номер столбца (соответствует x в графическом режиме).

В случае ошибки функции *ncurses* обычно возвращают константу ERR. Если функция не должна возвращать какое-то информативное значение (как, например, функция getch()), в случае успешного выполнения она возвращает константу OK.

Первая программа ncurses

Написание первой программы *ncurses* (она называется *cursed*, исходный текст вы найдете на DVD в файле **cursed.c**) мы начнем с перечисления заголовочных файлов.

```
#include <termios.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <stdlib.h>
#include <curses.h>
```

Помимо уже знакомых нам заголовочных файлов, в программу включен **curses.h**, который содержит объявления функций, констант и структур данных, экспортируемых библиотекой *ncurses*.

Прежде чем переходить к программированию *ncurses*, следует рассмотреть решение одной задачи, с которой сталкиваются все разработчики, использующие эту библиотеку. Речь идет об изменении размеров окна терминала (под размерами окна в данном случае понимается число строк и столбцов). Пользователи настоящих текстовых терминалов редко переключали их режимы, и готовы были мириться с последствиями своих действий. В наши дни, когда экраном терминала зачастую служит окно графической программы, пользователь вправе ожидать, что при изменении размеров окна работа консольной программы не нарушится, а ее интерфейс не развалится.

Когда размеры окна терминала меняются, выполняющаяся в нем программа получает сигнал SIGWINCH. Это одновременно и хорошо, и плохо. Хорошо – потому, что терминал информирует программу об изменении своих размеров, плохо – потому, что сигналы имеют особенность вмешиваться в работу программы. Например, если вы напишете программу, использующую *ncurses*, и не позаботитесь об обработке сигнала SIGWINCH, при изменении размеров окна терминала ваша программа может неожиданно завершиться, оставив терминал в ненормальном состоянии. Давайте посмотрим, как обрабатывается сигнал SIGWINCH в программе *cursed*.

```
void sig_winch(int signo)
{
    struct winsize size;
    ioctl(fileno(stdout), TIOCGWINSZ, (char *) &size);
    resizeterm(size.ws_row, size.ws_col);
}
```

Функция sig_winch() представляет собой обработчик сигнала SIGWINCH. Следует отметить, что изменение размеров экрана во время работы программы *ncurses* представляет собой довольно нетривиальную задачу и стандартного рецепта, описывающего, что нужно делать, не существует. Разработчики *ncurses*, как могли, постарались упростить решение этой задачи для программистов, введя функцию resizeterm(). Функцию resizeterm() следует вызывать сразу после изменения размеров окна терминала. Аргументами функции должны быть новые размеры экрана, заданные в строках и столбцах. Функция resizeterm() старается сохранить внешний вид и порядок работы приложения в окне терминала с новыми размерами, но это ей удается не всегда, с чем мы и столкнемся ниже. Необходимые для resizeterm() значения размеров окна мы получаем с помощью специального вызова ioctl(). При этом первым параметром функции ioctl() должен быть дескриптор файла устройства, представляющего терминал. Вторым параметром ioctl() должна быть константа TIOCGWINSZ, а третьим – адрес структуры struct winsize. Структура winsize, определенная в файле **<sys/ioctl.h>**, включает в себя поля ws_row и ws_col, в которых возвращается число строк и столбцов окна терминала.

Перейдем теперь к функции main() программы *cursed*:

```
int main(int argc, char ** argv)
{
    initscr();
    signal(SIGWINCH, sig_winch);
    noecho();
    curs_set(0);
    attron(A_BOLD);
    move(5, 15);
    printw("Hello, brave new curses world!\n");
    attroff(A_BOLD);
    attron(A_BLINK);
    move(7, 16);
    printw("Press any key to continue...");
    refresh();
    getch();
    endwin();
    exit(EXIT_SUCCESS);
}
```

Работа с *ncurses* начинается с вызова функции initscr(). Эта функции инициализирует структуры данных *ncurses* и переводит терминал в нужный режим. По окончании работы с *ncurses* следует вызвать функцию endwin(), которая восстанавливает то состояние, в котором терминал находился до инициализации *ncurses*.

После вызова initscr() мы устанавливаем обработчик сигнал SIGWINCH. Устанавливать обработчик SIGWINCH следует именно после инициализации *ncurses*, поскольку в нем используется функция resizeterm(), предполагающая, что библиотека *ncurses* уже инициализирована. Функция noecho() отключает отображение символов, вводимых с клавиатуры. Функция cur_set() управляет видимостью курсора. Если вызвать эту функцию с параметром 0, курсор станет невидимым, вызов же функции с ненулевым параметром снова «включит» его.

Функция attron() позволяет указать некоторые дополнительные атрибуты выводимого текста. Ей можно передать одну или несколько констант, обозначающих атрибуты (в последнем случае их следует объединить с помощью операции «|»). Например, атрибут A_UNDERLINE включает подчеркивание текста, атрибут A_REVERSE меняет местами цвет фона и текста, атрибут A_BLINK делает текст мигающим, атрибут A_DIM снижает яркость текста по сравнению с нормальной, атрибут A_BOLD делает текст жирным в монохромном режиме и управляет яркостью цвета в цветном режиме работы монитора. Специальный атрибут COLOR_PAIR() применяется для установки цветов фона и текста. На man-странице, посвященной функции attron(), вы найдете описания и других атрибутов. Все перечисленные выше атрибуты оказывают воздействие только на тот текст, который выводится после их установки.

Сбросить атрибуты можно с помощью функции attroff(). Так же, как и в случае с attron(), функции attroff() можно передать несколько »

констант, обозначающих атрибуты, разделенных символом «|». Сброс атрибута, как и его установка, влияет только на текст, напечатанный после сброса (текст, выведеный ранее с установленным атрибутом, остается без изменений). В нашей программе мы сначала устанавливаем атрибут A_BOLD. Теперь, до тех пор, пока мы не сбросим его, весь текст будет печататься жирным шрифтом. Но прежде, чем вывести текст этим шрифтом, мы используем еще одну возможность *ncurses* – вывод текста в произвольной области экрана. Функция move() устанавливает позицию курсора в окне stdscr. Первый аргумент функции – строка, второй аргумент – столбец, в котором должен находиться курсор. Последующий текст печатается, начиная с позиции курсора. Если попытаться поместить курсор за пределы окна, функция move() не станет выполнять никаких действий, и курсор останется на прежнем месте. Мы переводим курсор в позицию (5, 15) и выводим на экран строку "*Hello, brave new curses world!*" с помощью функции printw(). Функция printw() представляет собой аналог printf() для окна stdscr и имеет тот же список параметров, что и printf().

Затем мы сбрасываем атрибут A_BOLD с помощью функции attroff(), устанавливаем атрибут A_BLINK, переводим курсор в позицию (7,16) и распечатываем строку "*Press any key to continue...*".

Хотя мы уже напечатали две строки, на экране терминала все еще ничего нет. Для того чтобы выведенные нами символы стали видимыми, необходимо вызывать функцию refresh(). Функция refresh() является, в некотором роде, избыточной (действительно, почему бы не отображать распечатанный текст сразу же после вызова printw()?). Фактически, она представляет собой пережиток тех времен, когда терминал связывался с компьютером при помощи модема. Контролируя частоту вызовов refresh(), можно было сократить трафик между терминалом и компьютером.

В результате всех проделанных операций на экране, в заданных позициях, появятся две строки (**Рис. 1**) – одна выделенная жирным шрифтом, другая – мигающим (к сожалению, мигание на рисунке в журнале не изобразить).

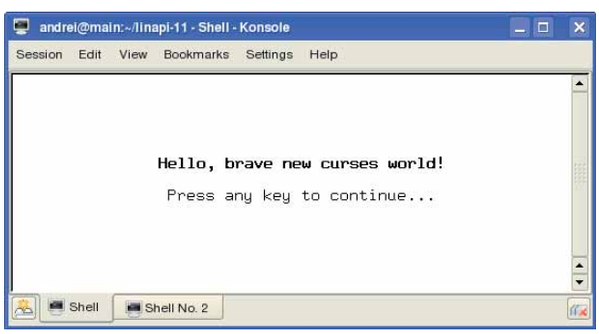


Рис. 1. Две строки от ncurses.

Функция getch(), которую мы вызываем далее, предназначена для считывания символов из потока ввода терминала. Функция считывает по одному символу и может работать в двух режимах: блокирующем (по умолчанию) и неблокирующем. В блокирующем режиме выполнение программы приостанавливается до появления символа в потоке ввода, а в неблокирующем – возвращает значение сразу же, независимо от того, есть ли символ в потоке ввода или нет (если символа в потоке ввода нет, функция getch() в неблокирующем режиме возвращает значение ERR). В режиме cbreak() (о котором будет подробнее рассказано в следующей статье), функция, считавшая символ, передает его программе, не дожидаясь, пока пользователь нажмет Enter. Таким образом, программа *cursed* завершается сразу же после нажатия на любую клавишу.

Библиотека *ncurses* не является частью стандартной библиотеки *glibc*, поэтому во время сборки программы ее нужно подключать явным образом. Например, чтобы скомпилировать *cursed*, следует набрать:

```
gcc cursed.c -o cursed -lncurses
```

Окна

В текстовых интерфейсах, построенных на основе *ncurses*, окна играют такую же важную роль, как и в графических интерфейсах. Прежде чем переходить к созданию приложений, использующих окна, необходимо внести некоторые уточнения в описание интерфейса *ncurses*. Прежде всего, вы должны понимать, что при работе с *ncurses* вы всегда имеете дело с окнами. В рассмотренной выше программе cursed мы работали с окном stdscr. Многие из функций *ncurses*, с которыми мы работали в программе cursed (attron(), move(), printw(), attroff(), getch()) являются частными вариантами функций, предназначенными специально для работы с окном *stdscr*. У этих функций есть обобщенные аналоги, способные работать с любым окном. Списки параметров обобщенных функций совпадают со списками параметров специальных функций, за исключением того, что первым параметром обобщенной функции должен быть указатель на структуру, определяющую окно. Например, для установки атрибутов текста в окне применяется функция wattron(). Первым параметром этой функции служит указатель на структуру, определяющую окно, а второй параметр wattron() полностью аналогичен параметру функции attron().

Как получить указатель на структуру, определяющую окно? Переменная *stdscr*, которую предоставляет библиотека *ncurses*, является указателем на структуру, представляющую корневое окно *stdscr*, занимающее весь экран. Эта переменная определена как

```
extern WINDOW * stdscr;
```

Тип WINDOW как раз и является структурой, описывающей окно. Из того, что *stdscr* является обычным окном *ncurses*, следует, что вместо функций, предназначенных специально для *stdscr*, мы можем использовать их обобщенные аналоги, указывая переменную *stdscr* в качестве идентификатора окна. Например, вызов attron(A_BOLD) эквивалентен вызову wattron(stdscr, A_BOLD). Обобщенным вариантом функции attroff() является функция wattroff(), а обобщенным вариантом функции move() функция wmove(). Вызов

```
move(5, 15);
```

из программы cursed можно заменить вызовом

```
wmove(stdscr, 5, 15);
```

Функции printw() соответствует обобщенная функция wprintw(). Функции getch() соответствует функция wgetch(), аргументом которой должен быть все тот же указатель на WINDOW.

Отметим, что многообразие функций ввода/вывода символов *ncurses* не исчерпывается парами getch()/wgetch() и printw()/wprintw(). В описании API библиотеки вы найдете множество других полезных возможностей.

Перейдем, наконец, к созданию собственных окон. Смысл создания окна заключается в том, чтобы ограничить область вывода текста (и область применения различных атрибутов) отдельными участками экрана. Библиотека *ncurses* предоставляет в наше распоряжение несколько функций, создающих новые окна. Самой простой и часто используемой является функция newwin(). Она принимает четыре параметра. Первые два соответствуют количеству строк и столбцов в создаваемом окне, а вторые указывают положение верхнего левого угла нового окна (строка и столбец) относительно окна *stdscr*. Функция newwin() возвращает указатель на структуру WINDOW или NULL в случае ошибки. После завершения работы с окном, выделенные ему ресурсы следует высвободить с помощью функции delwin(). Единственный параметр этой функции – указатель на структуру WINDOW, которую следует удалить.

Помимо функции newwin(), нам будет полезно познакомиться еще с двумя функциями: subwin() и derwin(). Эти две функции предназначены для создания подокон. Списки параметров у этих функций такие же, как и у newwin(), с той лишь разницей, что первым параметром каждой функции является указатель на структуру WINDOW, соответствующую родительскому окну. Последние два аргумента у subwin() и derwin() интерпретируются по-разному. У функции subwin() они задают положение верхнего левого угла окна относительно экрана, а у функции derwin() – относительно родительского окна.

Чем же подокно отличается от обычного окна? Окно и его подокно разделяют массив, в котором хранятся символы и их атрибуты. Новое подокно наследует все атрибуты своего родителя. Эти атрибуты затем могут быть изменены, что не повлияет на атрибуты родительского окна.

Мы займемся созданием окон в программе *cursedwindows* (файл **cursedwindows.c** на диске). На всякий случай заявляю, что у меня нет ни малейшего желания оскорбить Microsoft. Список заголовочных файлов и обработчик сигнала SIGWINCH у программы *cursedwindows* такие же, как и у программы *cursed*, так что в листинге мы их пропустим и рассмотрим только функцию main().

```
int main(int argc, char ** argv)
{
    WINDOW * wnd;
    WINDOW * subwnd;
    initscr();
    signal(SIGWINCH, sig_winch);
    curs_set(0);
    refresh();
    wnd = newwin(6, 18, 2, 4);
    box(wnd, 'l', '-');
    subwnd = derwin(wnd, 4, 16, 1, 1);
    wprintw(subwnd, "Hello, brave new curses world!\n");
    wrefresh(wnd);
    delwin(subwnd);
    delwin(wnd);
    move(9, 0);
    printw("Press any key to continue...");
    refresh();
    getch();
    endwin();
    exit(EXIT_SUCCESS);
}
```

В функции main() мы, как и прежде, инициализируем *ncurses* с помощью функции initscr() и устанавливаем обработчик SIGWINCH. Далее, как и в предыдущем примере, мы делаем курсор невидимым. После этого мы обновляем экран с помощью refresh(), а затем – создаем окно wnd с помощью функции newwin(). Оно насчитывает 6 строк и 18 столбцов, а его верхний левый угол находится в ячейке (2, 4) окна *stdscr*. Функция box(), которую мы вызываем далее, позволяет создать рамку вдоль границы окна. Ее аргументы: идентификатор окна и символы, используемые, соответственно, для рисования вертикальной и горизонтальной границы. Теперь было бы логично вывести в окно, украшенное рамкой, но тут возникает одна сложность. Поскольку символы рамки сами находятся внутри окна, символы текста могут затереть их в процессе вывода. Мы решаем эту проблему с помощью создания подокна subwnd внутри окна wnd и вывода текста в это подокно. Поскольку окно subwnd по размерам меньше, чем окно wnd, символы рамки не будут затерты.

Теперь мы можем распечатать текст – делается это с помощью функции wprintw(), которой передается идентификатор окна subwnd. Для того, чтобы символы, напечатаны в окне, стали видимыми, мы должны воспользоваться функцией wrefresh(). Мы вызываем ее только для окна wnd, поскольку именно оно содержит символьный массив и окно subwnd. Обратите внимание, что символы строки "*Hello, brave new curses world!*", которую мы печатаем в окне с помощью функции wprintw(), переносятся при достижении границы окна (**Рис. 2**). После завершения работы с окнами следует удалить структуры wnd и subwnd, воспользовавшись функциями delwin(). Весь вывод, выполненный в окне wnd, останется на экране (точнее, в окне *curscr*) до тех пор, пока вы не перезапишете его другим выводом.

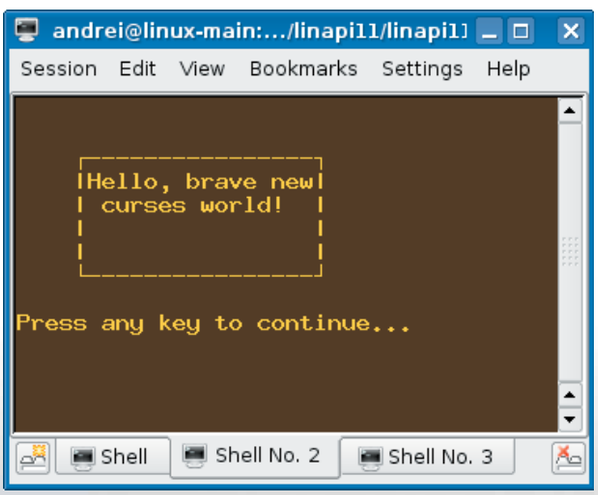


Рис. 2. Вывод текста в окне и за его пределами.

На этом наше знакомство с *ncurses* не заканчивается. В следующей статье мы рассмотрим управление цветом и ввод данных средствами *ncurses*. LXF

Через месяц Мы реализуем поддержку мыши и заставим наше приложение сиять всеми цветами радуги.

Сессии

ЧАСТЬ 3 Каждый линуксоид знает, что система уровня предприятия должна обеспечивать разграничение доступа. **Антон Черноусов** расскажет, что может предложить здесь Java EE.

В прошлый раз вы, под чутким руководством Александра, разобрались с JSP и переписали «Телефонную книгу» с учетом этой технологии. Сегодня мы двинемся дальше и поговорим об авторизации, сессиях, фильтрах и использовании разделяемых объектов в web-приложениях.

Процесс передачи информации

Давайте рассмотрим стандартную ситуацию: пользователь вызывает различные страницы одного web-приложения. Если бы этот «диалог» происходил по телефону, он был бы примерно таким:

Пользователь (набирает номер): Алло, это приложение?

Приложение: Да, это приложение.

(Пользователь кладет трубку)

Пользователь (набирает номер): Меня зовут Георг.

Приложение: Да, Георг, мы вас внимательно слушаем.

(Пользователь кладет трубку)

Пользователь (набирает номер): Дайте мне, пожалуйста, всю информацию.

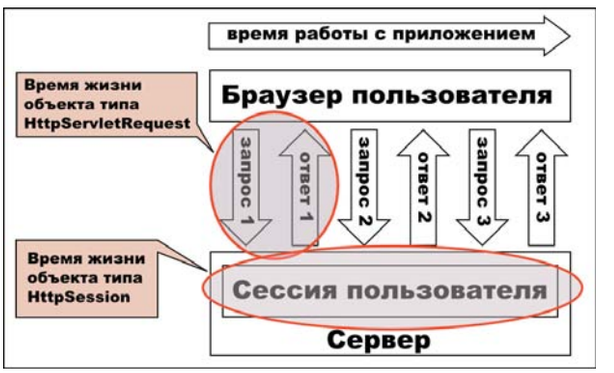
Приложение: Возьмите…

Отметим, что после каждого обмена репликами пользователь кладет трубку, разрывая соединение – именно таким образом в большинстве случаев и происходит обмен данными между браузером пользователя и сервером, на котором работает web-приложение. Возникает резонный вопрос: как передавать информацию между соединениями, открываемые в рамках одной сессии? Для этого могут использоваться самые различные методы:

- перезапись URI;
- скрытые поля;
- cookie;
- сессионный объект.

Из всех представленных методов наиболее простым и в тоже время мощным является сессионный объект (Session), реализованный в Java посредством интерфейса javax.servlet.http.HttpSession.

Ранее, чтобы получить данные от пользователя или обеспечить возможность их транспортировки внутри приложения, мы использовали объект типа HttpServletRequest. Основное отличие HttpSession заключается во времени жизни, схематично изображенном на **Рис.1**. Объект типа HttpServletRequest предназначен для передачи запроса (и необходимых данных) от браузера к web-приложению и существует только между запросом и ответом, в то время как объект HttpSession обеспечивает средства для хранения и доступа к данным пользователя на протяжении всего периода работы с приложением.



(Рис. 1) Время жизни объектов *HttpServletRequest* и *HttpSession*.

Следует, однако, иметь в виду, что отследить момент, когда пользователь перестает работать с приложением, не всегда возможно. Поэтому в настройках сервера устанавливается некоторое предельное время существования объекта HttpSession после получения последнего запроса.

Сессия

Объект session в JSP является предопределенным, то есть с ним можно сразу же начинать работать. Чтобы создать сессионный объект в сервлете, воспользуемся следующим методом:

```
HttpSession session = aRequest.getSession(true);
```

Здесь aRequest – это экземпляр объекта HttpServletRequest, то есть запрос переданный сервлету.

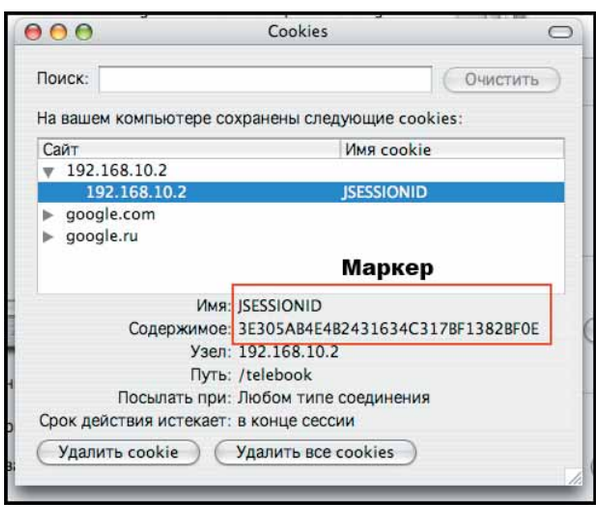
Классы, реализующие интерфейс HttpSession, имеют два замечательных метода: setAttribute(String, Object) и getAttribute(String). Метод setAttribute(String, Object) применяется для добавления объекта в сессию, а getAttribute(String) – для извлечения объекта из нее. Например:

```
session.setAttribute("sameKey",sameObject);
SameObject aObject = (SameObject) session.getAttribute("sameKey");
```

Месяц назад Мы познакомились с технологией JSP и отделили логику от дизайна.

и фильтры

Фактически, объект session – это таблица Hashtable, в которой можно хранить любое количество пар типа «ключ – объект». При использовании сессионного объекта данные приложения не отправляются пользователю так, как это происходит с cookie. Сессионный объект хранится на сервере персонально для каждого клиента. Сервер различает сессии с помощью маркера, который передается пользователю. Маркер хранится в cookies браузера до конца сессии, что накладывает некоторые ограничения на клиентское рабочее место (использование cookies для вашего приложения должно быть разрешено). В этом можно легко убедиться, просмотрев сохраненные cookies в вашем любимом браузере.



› (Рис. 2) Маркер сессии. Герои романа «Лабиринт отражений» пытались повесить друг другу маркер под видом невинного поцелуя, а наше web-приложение действует еще хитрее и незаметнее.

Авторизация

Логично предположить, что чаще всего сессии применяются там, где необходимо авторизовать пользователя и предоставлять доступ к определенным функциям приложения только при наличии соответствующих привилегий. Давайте обратимся к нашей телефонной книге: ее просмотр разрешен всем желающим, а для добавления нового телефона, редактирования и удаления записи необходима авторизация.

Давайте создадим JSP-страницу **auth.jsp**, которая будет запрашивать у пользователя имя и пароль. Сохраните файл в каталоге **jsps** и введите в него следующий текст:

```
<span style=”color: green;”><%=request.getAttribute(“message”)%>
</span>
<form action=”<%=request.getContextPath()%>/auth” method=”post”>
        <table>
            <tr><td>Логин: </td><td><input type=”text” name=”login”/>
</td></tr>
            <tr><td>Пароль: </td><td>
                    <input type=”password” name=”password”/></td>
</tr>
            <tr><td colspan=”2” align=”center”>
                    <input type=”submit” value=”Авторизоваться”/></td>
</tr>
        </table>
</form>
```

Легко видеть, что в форме имеется два поля ввода и одна кнопка. Заметим также, что поле для ввода пароля имеет тип password, так что вместо нажатых клавиш в окне браузера будут отображаться до боли знакомые «звездочки» (‘*’).

Теперь внесем изменения в AddressBookServlet: расширим ветвление в методе handle следующим образом:

```
}else if (“/auth”.equals(target)) {
        handleAuth(aRequest, aResponse);
    }
```

Дело за малым – осталось написать метод handleAuth(aRequest, aResponse), который и будет обрабатывать данные, введенные в форму, то есть производить авторизацию пользователя.

```
  private void handleAuth(HttpServletRequest aRequest,
            HttpServletResponse aResponse) throws IOException,
ServletException {
      String login = aRequest.getParameter(“login”);
      String password = aRequest.getParameter(“password”);
      if ((login != null) && (password != null)) {
        // here is auth process
        if ((login.equals(“user”)) && (password.equals(“userPass”))) {
          //here is success auth
          HttpSession session = aRequest.getSession(true);
          session.setAttribute(“auth”,aRequest.getParameter(“login”));
          aRequest.setAttribute(“message”, null);
          outputPage(“index.jsp”, aRequest, aResponse);
        } else {
          //here is failed auth
          aRequest.setAttribute(“message”,
                    “Неверный логин или пароль, повторите ввод
данных”);
          outputPage(“auth.jsp”, aRequest, aResponse);
        }
      } else {
        //here is no data to auth
        aRequest.setAttribute(“message”,
                “Логин и пароль не могут быть пустыми, повторите
ввод данных”);
        outputPage(“auth.jsp”, aRequest, aResponse);
```

```
    }
  }
```

Чтобы не усложнять приложение, мы используем простейший способ авторизации – имя пользователя и пароль жестко зашиты в теле метода. С точки зрения безопасности и масштабируемости лучше использовать для хранения этих данных БД.

В случае удачной авторизации мы помещаем имя пользователя в объект session и перенаправляем его на главную страницу. Если же авторизация не удалась (введено неверное имя пользователя и/или пароль), пользователю будет предложено повторить попытку.

Таким образом, если авторизация прошла успешно, в сессии пользователя будет сохранен объект, содержащий его имя. Используя этот факт, можно изменить **index.jsp** и спрятать ссылку на добавление нового телефона от посторонних глаз:

```
<%String name = (String)session.getAttribute("auth");
 if (name != null){ %>
    <p>Вы авторизованны как: <%=name%> </p>
    <a href="<%=request.getContextPath()%>/add">Добавить
запись</a><br/>
 <%} else {%>
    <a href="<%=request.getContextPath()%>/auth">
Авторизоваться</a><br/>
 <%} %>
```

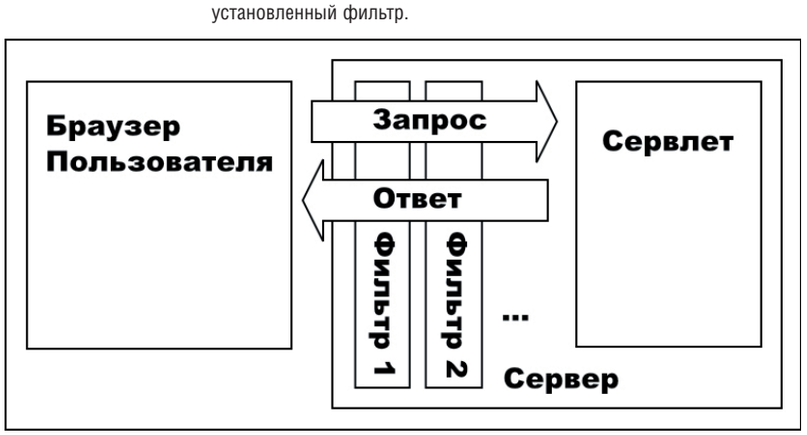
Мы приветствуем пользователя, используя значение атрибута auth и предоставляем ему доступ к функции добавления нового контакта. Чтобы спрятать удаление и редактирование записей от неавторизованного пользователя, необходимо внести аналогичные изменения в файл **view.jsp**.

Фильтры

Наше приложение, к сожалению, страдает проблемами безопасности: злонамеренный пользователь может обойти процедуру авторизации, обратившись к функциям редактирования/удаления напрямую, по адресу request.getContextPath() + действие (/add; /edit; /remove); можно также непосредственно вызывать JSP-страницы, расположенные в каталоге **/jsps** – нужно только узнать их имена.

Что же делать? Проверять, авторизован ли пользователь перед выполнением привилегированного действия? Это неудобно – если количество JSP будет расти, это заставит вас написать много строк однотипного кода, поддержание которых будет отнимать ваше драгоценное время. Мы пойдем другим путем и воспользуемся так называемыми фильтрами.

Схема работы фильтров представлена на **Рис. 2**. Если для адреса, на который отображен сервлет, осуществляется фильтрация, запрос будет передан сервлету только после того, как пройдет через каждый установленный фильтр.



› (Рис. 3) Схема прохождения запроса через фильтры.

На самом деле, фильтр представляет из себя обыкновенный Java-класс, реализующий интерфейс javax.servlet.Filter. Например:

```
public class FirstFilter implements Filter {
  private FilterConfig filterConfig;
  public void init(FilterConfig filterConfig) throws ServletException {
    System.out.println("Filter init");
    this.filterConfig = filterConfig;
  }

  public void doFilter(ServletRequest aRequest, ServletResponse
aResponse, FilterChain filterChain)
  throws IOException, ServletException {
    System.out.println("Filter used");
    filterChain.doFilter(aRequest, aResponse);
  }

  public void destroy() {
    System.out.println("Filter dead");
    this.filterConfig = null;
  }
}
```

Реализовав всего три метода: init (инициализация фильтра в момент старта приложения), destroy (освобождение ресурсов перед завершением работы приложения) и doFilter (собственно метод, выполняющий фильтрацию), вы получаете класс, способный существенным образом повлиять на работу вашего приложения.

Обратите внимание, что в приведенном выше примере метод doFilter заканчивается вызовом метода doFilter(aRequest, aResponse) объекта filterChain – это обеспечивает вызов следующего фильтра в цепочке (естественно, если фильтров несколько). Если фильтров больше нет, то управление будет передано следующему ресурсу, например, сервлету.

Остановимся подробнее на методе init, а точнее на объекте, реализующем интерфейс FilterConfig. Этот объект имеет четыре замечательных метода:

- getFilterName() – возвращает имя фильтра;
- getInitParameterNames() – возвращает объект Enumeration, который содержит в своем теле имена параметров текущего фильтра;
- getInitParameter(String) – возвращает значение параметра, имя которого было передано в качестве аргумента;
- getServletContext() – возвращает объект ServletContext, о котором мы поговорим ниже.

Как вы уже, наверное, поняли, объект FilterConfig позволяет получить доступ к конфигурации фильтра, которая была задана при его объявлении в дескрипторе развертывания (web.xml).

Ограничение доступа

Итак, воспользуемся фильтрами для ограничения доступа к ресурсам нашего приложения. Прежде всего, запретим пользователю обращаться напрямую к JSP, и, при попытке запросить ресурс из каталога **/jsps**, заставим его перейти на индексную (первую) страницу. Для этого напишем простой фильтр AccessFilter. Метод doFilter будет выглядеть следующим образом:

```
  public void doFilter(ServletRequest aRequest,
      ServletResponse aResponse, FilterChain filterChain)
      throws IOException, ServletException {
        ((HttpServletResponse)aResponse).
       sendRedirect(((HttpServletRequest)aRequest).getContextPath());
  }
```

Благодаря методу sendRedirect, вместо ожидаемого ресурса неавторизованный пользователь увидит **index.jsp**. Чтобы фильтр заработал, его необходимо подключить в дескрипторе развертывания. Это делается при помощи следующих строк:

```
  <filter>
    <filter-name>AccessFilterName</filter-name>
    <filter-class>AccessFilter</filter-class>
  </filter>
  <filter-mapping>
```

```
    <filter-name>AccessFilterName</filter-name>
    <url-pattern>/jsps/*</url-pattern>
  </filter-mapping>
```

В секции filter мы объявляем, что имени фильтра AccessFilterName соответствует класс AccessFilter, а в секции filter-mapping указываем, на какие объекты распространяется зона действия фильтра. В данном случае фильтр работает по шаблону, то есть используется для всех адресов типа /jsps/*, где вместо звездочки может быть все, что угодно.

Теперь давайте создадим класс AuthFilter, который будет ограничивать доступ к некоторым действиям для неавторизованных пользователей. Метод doFilter будет выглядеть следующим образом:

```
  public void doFilter(ServletRequest aRequest,
      ServletResponse aResponse, FilterChain filterChain)
      throws IOException, ServletException {
    HttpSession session = ((HttpServletRequest) aRequest).
getSession();
    String name = (String) session.getAttribute("auth");
    if (name != null) {
      filterChain.doFilter(aRequest, aResponse);
    } else {
      aRequest.setAttribute("action", "auth");
      RequestDispatcher dispatcher = aRequest.getRequestDispatcher("/
auth");
      dispatcher.forward(aRequest, aResponse);
    }
  }
```

Если в текущей сессии не задан атрибут auth, пользователь будет отправлен на страничку авторизации. Добавим в дескриптор развертывания следующие строки:

```
  <filter>
    <filter-name>AuthEditFilter</filter-name>
    <filter-class>AuthFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>AuthEditFilter</filter-name>
    <url-pattern>/edit</url-pattern>
  </filter-mapping>
  <filter>
    <filter-name>AuthAddFilter</filter-name>
    <filter-class>AuthFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>AuthAddFilter</filter-name>
    <url-pattern>/add</url-pattern>
  </filter-mapping>
  <filter>
    <filter-name>AuthRemFilter</filter-name>
    <filter-class>AuthFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>AuthRemFilter</filter-name>
    <url-pattern>/remove</url-pattern>
  </filter-mapping>
```

Обратите внимание, что хотя наш фильтр реализован одним-единственным классом, он может быть доступен по нескольким именам (в нашем случае: AuthEditFilter, AuthAddFilter, AuthRemFilter) и работать для разных URL.

Чтобы закрыть тему развертывания фильтров, рассмотрим пример настройки параметров фильтра в дескрипторе:

```
    <init-param>
      <description>
        That is description
      </description>
      <param-name>
        sameParamName
      </param-name>
      <param-value>
        sameParamValue
      </param-value>
    </init-param>
```

Секцию init-param необходимо создавать отдельно для каждого параметра, при этом она должна быть размещена внутри секции filter. Обязательными являются подсекции param-name и param-value, которые задают название параметра и его значение, соответственно.

Доступ к общим объектам

Наше приложение уже вполне работоспособно, однако рано или поздно нам захочется расширить его функциональность, и, возможно, для этого потребуется уже не один сервлет, а несколько, причем все они будут обращаться к некому общему ресурсу (информации или объектам). Например, вы можете захотеть узнать, сколько пользователей в данный момент авторизовано в приложении или получить из двух несвязанных сервлетов доступ к информации, хранящейся в одном файле.

Подобно сессионным объектам, существует и контекстный объект, который обеспечивает доступ к общим ресурсам из разных мест приложения. Использование контекстного объекта очень похоже на использование сессионного, и получить его экземпляр в сервлете можно следующим образом:

```
ServletContext sc = this.getServletContext();
```

Этот объект обладает следующими методами:

- getAttribute(String) – получение общего объекта по ключу;
- getAttributeNames() – получение списка ключей общих объектов;
- setAttribute(String, Object) – добавление нового объекта и соответствующего ему ключа;
- removeAttribute(String) – удаление объекта, соответствующего ключу, из списка общедоступных объектов;

Несмотря на то, что в этой статье контекстному объекту уделено немного внимания, он является одним из мощнейших инструментов для обеспечения работы web-приложения.

Вместо заключения

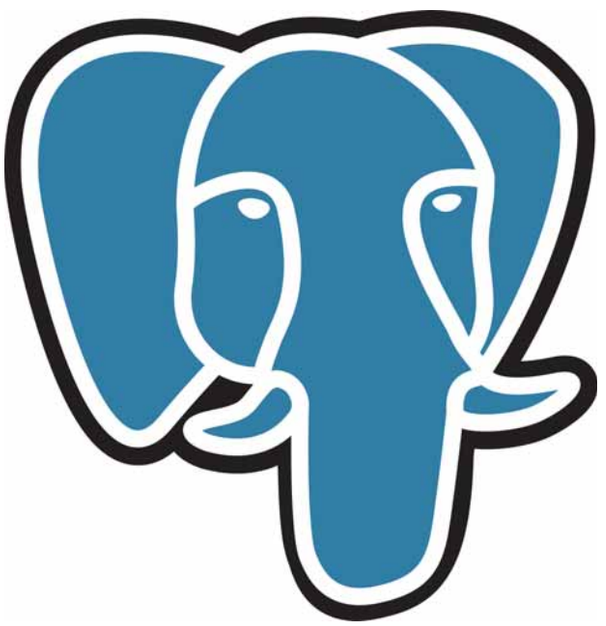
Сегодня мы поговорили о сессионных и контекстных объектах, узнали как создавать и использовать фильтры, рассмотрели простейший пример авторизации пользователя (конечно, надо сделать оговорку, что было представлено весьма небезопасное и не промышленное решение, хотя для простого офисного приложения его может быть и достаточно). Наконец, мы модифицировали наше web-приложение «Телефонная книга», применяя практически все изученные возможности.

В следующий раз мы рассмотрим паттерн MVC и его вариацию для создания web-приложений: Model2. LXF

Через месяц Что общего у SmallTalk и Java, кроме объектов? Мы займемся изучением паттерна Model-View-Controller..

Дополнительные

ЧАСТЬ 6 В заключительной статье цикла **Евгений Балдин** хочет остановиться на моментах, с которыми стоит разобраться поглубже.



Устреми свои мысли на высшее Я, свободный от вожделения и себялюбия, исцелившись от душевной горячки, сражайтесь, Арджуна!

Зеркало. Понедельник начинается в субботу.

Все рассказать и предусмотреть невозможно, хотя бы по той простой причине, что составляя планы, мы изменяем реальность. Измененная реальность, в свою очередь, требует измененных планов, и так – до бесконечности. Исходя из этой теории, мы не будем пытаться рассказать о *PostgreSQL* все, не откажем себе в удовольствии раскрыть напоследок некоторые из его особенностей чуть более подробно.

Резервное копирование

Если уж вы завели хранилище информации, то его надо беречь. При этом навязчивая боязнь порчи данных, переходящая в манию, является обязательной чертой характера хорошего администратора базы данных. Только в этом случае можно подстелить соломки в нужном месте, до того как поскользнуться. То, что при этом весь дом будет покрыт высушенной травой – обычные издержки производства.

pg_dump/pg_restore

Просто копировать физические файлы базы данных – не самый лучший способ резервирования, потому что на время выполнения этой операции придется как минимум остановить сервер. Для создания консистентной копии базы данных проще всего воспользоваться программой pg_dump (man pg_dump), которая работает как обычный клиент:

```
> pg_dump -Fc «база данных» > «файл резервной копии»
```

Опция -Fc определяет формат резервной копии: custom. В этом случае сохраняются не только SQL-структуры, но и большие объекты (lobj).

Для восстановления базы данных из ее резервной копии используется зеркальная процедура pg_restore (man pg_restore):

```
> pg_restore -d «новая база данных» «файл резервной копии»
```

Используя pg_restore с опцией -l, можно получить список всех таблиц, находящихся в резервной копии, а с помощью опции -L – указать список таблиц, которые надо восстановить. Иногда требуется лишь частичное восстановление данных, например, для отката одной конкретной таблицы.

Так как pg_dump и pg_restore сконструированы с прицелом на работу в конвейере, то их удобно использовать в скриптах. Резервная копия представляет собой, в основном, ASCII-файл, поэтому при формировании резервной копии/восстановления имеет смысл предусмотреть фильтр для сжатия данных, например, bzip2.

При восстановлении больших объектов (lobj) очень важно, чтобы pg_restore отработал без ошибок от начала и до конца. Причина этого в том, что при восстановлении больших объектов создается временная таблица, где хранится перекодировка из старой нумерации OIDов в новую. Если в процессе восстановления произошел сбой, эта таблица теряется и ссылки на большие объекты в таблицах не обновляются. В результате большие объекты загружаются в базу данных, но ссылки на них отсутствуют. Это один из примеров того, как нестандартные расширения могут приводить к неудобствам.

Резервирование «нон-стоп»

pg_dump может создавать резервную копию, не особо мешая работе базы данных, так как это всего-навсего еще один клиент. В классическом подходе к резервированию есть одна неприятность: информация между моментом снятия копии и крахом базы данных теряется. Иногда это терпимо, так как подобное случается редко, но есть случаи, когда потерянные запросы означают потерянные деньги в полном смысле этого слова. И здесь на помощь приходит журналирование транзакций – своего рода находка для параноика.

Организация непрерывного резервирования – довольно сложная процедура, и для ее реализации следует обратиться к разделу документации, который так и называется «On-line backup and point-in-time

Месяц назад Настройка *PostgreSQL* под ваши нужды и сервер 1С:Предприятия на базе открытого ПО.

ГЛАВЫ

recovery (PITR)». В нем представлено пошаговое руководство к действию длиной чуть меньше пяти тысяч слов, что составляет около пятнадцати страниц формата A4.

Основная идея заключается в архивировании журнала транзакций. Формально, все действия *PostgreSQL* можно представить как последовательные записи в этом журнале. На диске журнал транзакций разбивается на независимые файлы или сегменты (segment files), размер которых по умолчанию равен 16 МБ. *PostgreSQL* можно настроить на копирование сегментов в место для хранения резервных копий (параметр archive_command в **postgresql.conf**). При этом нет необходимости хранить абсолютно все записи, достаточно оставлять только те, которые были сделаны после резервирования. Для локализации момента времени, которому соответствует та или иная резервная копия, используются хранимые процедуры **pg_start_backup/pg_stop_backup**.

При восстановлении можно вернуть не только последнее состояние базы данных, но и то, в котором она была на указанный момент времени (естественно, все ограничено объемом сохраненных сегментов). Таким образом, при желании можно организовать своеобразное путешествие в прошлое (**point-in-time recovery**).

Переезд на новую версию PostgreSQL

По умолчанию, при выполнении pg_dump на выходе получаются SQL-команды, так что для восстановления можно воспользоваться psql, указав файл резервной копии с помощью ключа -f. Таким образом, структура резервной копии зависит только от версии SQL, которую поддерживает данный сервер. Это позволяет достаточно легко обновлять *PostgreSQL*, даже в случае, когда представление данных внутри самого сервера изменяется.

Переезд с версии на версию гарантированно можно выполнить в четыре этапа:

1. сделать резервную копию с помощью pg_dumpall;
2. остановить старый сервер;
3. запустить новый сервер;
4. восстановить базу данных с помощью pg_restore или psql.

Если меняется только минорная версия *PostgreSQL* (последняя цифра в номере), то в принципе можно упустить этап (1) и (4). Но в любом случае не следует забывать о фобии потери данных. В принципе, можно исключить пункты (2) и (3), воспользовавшись конвейером:

```
> pg_dump -h host1 «БД» | psql -h host2 «БД»
```

host1 и host2 – компьютеры, «с которого» и «на который» переезжает база данных, соответственно.

Репликация слонов

База данных подразумевает централизацию: все складывается в одно место. Это может стать проблемой. Некоторые проблемы не решаются, но если требуется всего-на всего ускорить доступ на чтение, то репликация базы данных может оказаться спасением. Побочным эффектом репликации является повышение надежности, так как число консистентных копий данных увеличивается. Создание кластера баз данных – это глобальный инструмент для решения многих проблем, но и сложности в управлении таким кластером также будет предостаточно.

Для репликации *PostgreSQL* существует несколько решений, как закрытых[1], так и свободных. Самой популярной свободной системой репликации является *Slony I* (**http://slony.info/**). *Slony I* поддерживает репликацию **master/slaves**[2].

Список возможных преимуществ, которые можно получить, наладив репликацию, таков:

- Организуются дополнительные копии данных, которые никогда не бывают лишними (мы помним о благотворном влиянии паранойи).
- Разгружается центральный сервер: теперь он может заниматься действительно важными делами, не отвлекаясь на мелочи. Например, процедура полного резервирования довольно ресурсоемкая. Вполне можно поручить это задание одному из вспомогательных серверов. Аналогично, можно организовать сервер, который имеет очень длинный журнал транзакций, позволяющий (в случае необходимости) откатиться максимально далеко в прошлое.
- Можно перенести вспомогательный сервер поближе к клиенту, чтобы сэкономить время, которое уходит на подключение к базе данных,
- Дополнительные сервера позволяют-таки получить доступ к данным, даже если связь с центральным сервером полностью потеряна.

Вспомогательные сервера вовсе не обязаны получать обновления непосредственно с главного сервера (Master to multiple cascades Slaves). Любой сервер, который получает данные из надежного источника, может быть сконфигурирован так, чтобы рассылать их далее по цепочке. Данная особенность позволяет легко масштабировать всю систему. Развернуть и запустить репликацию можно не останавливая центральный сервер.

Для привязки к событиям INSERT/DELETE/UPDATE используются триггеры *PostgreSQL*. Выполнение действий реализуется через хранимые процедуры. Слежением за процессом репликации занимается системный демон *slon* – он должен быть запущен на каждом из узлов кластера. Администрирование осуществляется посредством командного процессора *slonik*.

Административная утилита *slonik* реализована как программа, ориентированная на выполнение в командной строке и в скриптах. Синтаксис команд, воспринимаемых *slonik*'ом, напоминает SQL. Команды следует передавать на STDIN. Перед выполнением запроса *slonik* анализирует синтаксис и в случае наличия проблем выдает сообщение об ошибке.

Подробно о настройке кластера можно прочитать в документации к пакету. На русском языке доступно пошаговое руководство, написанное Евгением Кузиным, правда, возможно уже устаревшее: **http://www.kuzin.net/work/sloniki-privet.html**. В случае возникновения проблем для начала следует поискать решение в стандартном FAQ: **http://linuxfinances.info/info/faq.html**. »

---

[1] Например, **http://www.commandprompt.com/products/mammothreplicator** – *Mammoth PostgreSQL + Replication*.

[2] Имя *Slony-II* зарезервировано для версии, которая будет поддерживать режим multi-master. На текущий момент перспективы этой версии довольно туманны. Организовать надежное решение для требуемого режима очень сложно в силу большого количества принципиальных проблем: **http://www.dbspecialists.com/presentations/mm_replication.html**. Для использования в режиме multi-master *PostgreSQL* поддерживает отложенные транзакции (two-phase commit). Two-phase commit реализуются с помощью SQL-запросов PREPARE TRANSACTION и COMMIT PREPARED.

Принципиальные ограничения

Большие объекты не реплицируются. Это происходит потому, что *Slony I* работает на триггерах, а операции с большими объектами триггерным механизмом не отлавливаются. Иными словами, реплицируются только таблицы и последовательности. Для того, чтобы репликация работала автоматически, лучше отказаться от использования больших объектов, благо существуют соответствующие бинарные типы данных, вполне годящиеся на замену.

На начало марта 2007 года последняя версия *Slony I* имела номер 1.2.2. Для ее работы необходим *PostgreSQL* старше 7.3.3, так как требуется обязательная поддержка пространства имен (namespace). При репликации предполагается, что все базы данных создавались с указанием одной и той же кодовой страницы[3] и текущая кодовая страница совпадает с ней. Задача временной синхронизации серверов выходит за рамки функционирования *Slon*'ов – для этого следует запустить специальную службу Network Time Protocol (http://www.ntp.org).

Процедуры изменения схемы базы данных (database schema, DDL – Язык определения данных), следует производить посредством передачи команд через *slonik* с помощью префикса EXECUTE SCRIPT. Это гарантирует, что, например, изменение числа столбцов в таблице произойдет во всем кластере до того, как в нее начнут добавляться данные.

» Побочным эффектом репликации является возможность использовать ее при обновлении сервера *PostgreSQL*. Это удобно, когда объем базы данных становится очень большим, а останавливать ее на время смены версии нежелательно.

Локаль

Локаль (locale) – это набор соглашений, специфических для отдельно взятого языка в отдельно взятой стране[4]. Локаль и кодовая страница базы данных выбираются при ее создании с помощью команды initdb:

```
> initdb --locale=ru_RU.UTF-8 --lc-numeric=POSIX
```

Результат выполнения SQL-запросов может зависеть от локали. Это, например, проявляется при сортировке текстовых данных или при выполнении функций **upper/lower/initcap**.

Для корректной работы базы данных с устанавливаемой локалью необходимо, чтобы данная локаль поддерживалось системой. Вывести список поддерживаемых локалей можно с помощью команды locale -a.

Так как локализация *PostgreSQL* проводилось Олегом Бартуновым, поддерживаются все русские кодовые страницы. При наборе русских текстов можно использовать следующие из них: KOI8 (aka KOI8R), WIN1251 (aka WIN), WIN866 (aka ALT), ISO_8859_5, UTF8 (aka Unicode) и MULE_INTERNAL[5].

Кодовая страница клиента может отличаться от кодовой страницы сервера. Например, установить кодовую страницу в сессии psql можно следующим образом:

```
mydb-> \encoding KOI8R
mydb-> show CLIENT_ENCODING;
 client_encoding
-----------------
 KOI8R
(1 запись)
```

При этом на самом деле используется функция PqsetClientEncoding() из библиотеки *libpq* (см. LXF'99), которая, в свою очередь, выполняет SQL-запрос SET CLIENT_ENCODING TO.

После установки кодовой страницы *PostgreSQL* выполняет автоматическое преобразование запросов между кодовыми страницами сервера и клиента, если это, конечно возможно. Для русских кодовых страниц все варианты преобразований имеются. При желании можно создать свою таблицу преобразования с помощью SQL-запроса CREATE CONVERSION.

VACUUM/ANALYZE

Администрируя *PostgreSQL*, следует помнить, что для его нормального функционирования следует регулярно «мыть руки» и «чистить зубы», то есть выполнять команды VACUUM и ANALYZE. Это необходимо, поскольку иначе не получится заново использовать дисковое пространство, занимаемое удаленными или измененными строками и не удастся обновить статистику для планировщика запросов. И то, и другое отрицательно сказывается на эффективности использования ресурсов и производительности.

Начиная с версии *PostgreSQL* 8.1, сервер может самостоятельно запускать еще один системный процесс, который, соответственно, так и называется: autovacuum daemon. Все настройки для этого процесса хранятся в **postgresql.conf.** К значениям этих параметров следует отнестись крайне внимательно.

Если по каким-то причинам демон было решено не запускать, то в любом случае придется производить сборку мусора и набор статистики вручную, с помощью команды vacuumdb (man vacuumdb):

```
> vacuumdb -ze
VACUUM ANALYZE;
VACUUM
```

Мониторинг активности базы

Текущую активность базы данных легко оценить с помощью простой команды ps:

```
> ps auxww | grep ^postgres
postgres … postmaster -i
postgres … postgres: writer process
postgres … postgres: stats buffer process
postgres … postgres: stats collector process
postgres … postgres: baldin mydbase [local] idle
```

Так как для каждого клиента создается своя копия процесса postmaster, это позволяет подсчитать число активных клиентов. Статусная строка дает информацию о состоянии клиента. Фразы writer process, stats buffer process и stats collector process соответствуют системным процессам, запущенным при старте самим *PostgreSQL*. Пользовательские процессы имеют статусную строку вида

```
postgres: «пользователь» «база» «хост» «статус»
```

«пользователь», «база» и «хост» соответствуют имени пользователя «пользователь» подключившегося к базе «база» с компьютера «хост». «статус» может принимать следующие значения:

- idle – ожидание команды от клиента;
- idle in transaction – ожидание команды от клиента внутри транзакции (между BEGIN и окончанием транзакции);
- SQL-команда – выполняется эта команда, например, SELECT;
- waiting – ожидание разблокирования таблицы, занятой другим процессом.

Если в **postgresql.conf** разрешен сбор статистики (опции stats_start_collector и stats_row_level), то информация об активности базы данных собирается в специальных системных таблицах. Ту же информацию, что доступна с помощью ps, можно извлечь из таблицы pg_stat_activity, а в pg_stat_all_tables лежат данные о числе обращений к каждой из таблиц базы. Подробнее обо всех имеющихся таблицах можно прочитать в главе «Viewing Collected Statistics» стандартной документации *PostgreSQL*. Собранная информация может оказаться полезной для оценки эффективности работы базы данных и выполнения запросов. Например, pg_stat_all_indexes поможет оценить эффективность и частоту использования индексов при реальной работе с базой. Подробную информация о блокировках можно почерпнуть в таблице pg_locks.

[3] Это замечание относится к ключу --encoding команды createdb.
[4] В общем случае говорить, что локаль определяется только страной, неправильно. Например, в Канаде могут быть определены две локали: язык «Канада/Английский» и язык «Канада/Французский». Аналогично язык «Великобритания/Английский» не эквивалентен языку «Соединенные Штаты/Английский».
[5] То, что используется в *emacs*.

log

Когда что-то работает в фоновом режиме, бывает полезно иметь обратную связь. Поэтому лучше, чтобы журнальный файл (log) существовал. Создавать его самостоятельно или воспользоваться службой *syslog* – зависит от обстоятельств. Следует только учитывать, что *syslog* на каждой записи производит операцию sync, что может серьезно замедлить доступ к диску на котором лежит журнальный файл [поведение *syslog* можно изменить, см. стр. 39, – *прим. ред.*].

Послесловие

На этом мы заканчиваем цикл статей о *PostgreSQL*. Но, я надеюсь, ваше взаимодействие с этим замечательным образчиком программного искусства будет только расширяться. С одной стороны, довольно странно, что одну программу пришлось рассматривать на протяжении целых шести глав, причем дальше обзора, как правило, зайти не удавалось. С другой стороны, задача сохранения и обеспечения доступа к уже имеющимся данным является одной из самых важных для выживания человечества (в буквальном смысле). Поэтому эта область деятельности в информатике является, пожалуй, самой развитой. Здесь теория переходит в практику.

Далеко не все было нами упомянуто, и далеко не все упомянутое было описано достаточно подробно. Та же тема полнотекстового поиска сама по себе может занять не одну главу. Картографические и астрофизические типы данных также имеют за собой фундаментальную базу и интересные сферы приложения. *PostgreSQL* – это хранилище, но за каждым типом данных стоит своя уникальная история и инструкция по использованию. Тема супербольших баз данных, где кластеризация является обязательной, суперсложна и не менее интересна. Переход к многопроцессорным архитектурам расширяет границы применимости баз данных. Я не сильно удивлюсь, если «угрозы» сделать из файловой системы специализированную базу данных воплотятся в обычной базе данных общего назначения, которая будет хранить в себе файловую систему или хотя бы пользовательские документы, где не особо важна скорость доступа, но естественный полнотекстовый поиск будет весьма кстати. В любом случае, в будущем без баз данных делать нечего, и *PostgreSQL* там будет наверняка. Присоединяйтесь. LXF

К вопросу о происхождении Слонов

В документации к пакету *Slony I* для англоязычной аудитории идет специальное разъяснение:

- **слон** – это русский elephant,
- множественная форма от слова «слон» – это «**слоны**»,
- **слоник** – это маленький elephant.

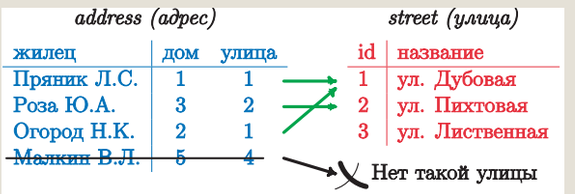
Термин *Slony I* – это реверанс в сторону Вадима Михеева, который создал прототип для системы репликации *rserv* на языке Perl. Проект был спонсирован фирмой Afilias (http://afilias.info), которая наняла для этого одного из основных разработчиков *PostgreSQL* Яна Вика (Jan Wieck). Со слов Яна, проект с самого начала планировался как программа с открытыми исходными текстами, которые всегда были доступны публично. Это яркий пример того, что коммерческие фирмы могут сделать весомый вклад в открытые разработки без каких-либо задних мыслей, как участники свободного сообщества.

Азбука SQL: Д

Ссылочная целостность

Если все значения столбца A присутствуют в столбце B, то говорят, что столбец A ссылается на столбец B. Если столбец B является потенциальным ключом (соответствует ограничению PRIMARY KEY или UNIQUE), то любое значение столбца A соотносится ровно одному значению столбца B. В этом случае столбец A является внешним ключом (foreign key) по отношению к столбцу B, который в свою очередь называется родительским ключом.

Реляционная база данных не должна содержать значений внешних ключей, не имеющих соответствия; иными словами, должна поддерживаться ссылочная целостность (referential integrity).



Столбец «улица» таблицы address является внешним ключом, ссылающимся на столбец «id» таблицы street. Ссылочная целостность не выполнена для Малкина В.Л., так как в таблице street нет улицы с идентификатором 4.

Ссылочная целостность поддерживается *PostgreSQL* посредством механизма внешних ключей. Для создания внешнего ключа необходима таблица с потенциальным ключом:

```
wikidb=> CREATE TABLE street ( -- список улиц
wikidb(>     id integer PRIMARY KEY, -- ключ
wikidb(>     name text -- название улицы
wikidb(> );
NOTICE:  CREATE TABLE / PRIMARY KEY will create
        implicit index “street_pkey” for table “street”
wikidb=> INSERT INTO street VALUES (1,’ул. Дубовая’);
wikidb=> INSERT INTO street VALUES (2,’ул. Пихтовая’);
wikidb=> INSERT INTO street VALUES (3,’ул. Лиственная’);
```

Теперь можно создать таблицу с внешним ключом:

```
wikidb=> CREATE TABLE address ( -- адресный список
wikidb(>     person text, -- имя человека
wikidb(>     house text, -- номер дома
wikidb(>     -- внешний ключ
wikidb(>     street_id integer REFERENCES street (id)
wikidb(> );
wikidb=> INSERT INTO address VALUES (‘Пряник Л.С.’,’1’,1);
wikidb=> INSERT INTO address VALUES (‘Роза Ю.А.’,’3’,2);
wikidb=> INSERT INTO address VALUES (‘Огород Н.К.’,’2’,1);
```

Нельзя добавить во внешний ключ данные, отсутствующие в родительском ключе:

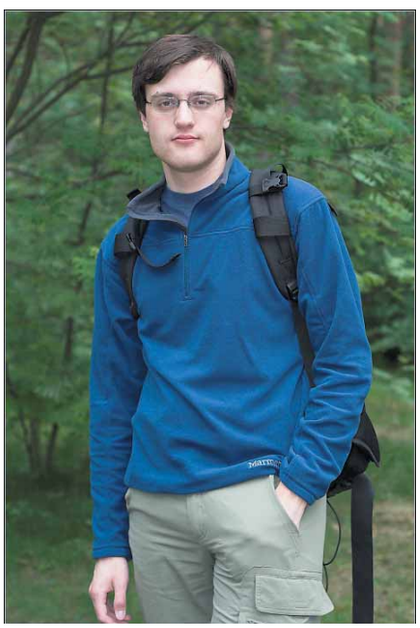
```
wikidb=> INSERT INTO address VALUES (‘Малкин В.Л.’,’5’,4);
ERROR:  insert or update on table “address” violates
      foreign key constraint “address_street_id_fkey”
DETAIL:  Key (street_id)=(4) is not present in table “street”.
```

При настройках по умолчанию, из родительского ключа нельзя удалить данные, на которые кто-то ссылается:

```
wikidb=> DELETE FROM street WHERE id=1;
ERROR:  update or delete on “street” violates foreign key
      constraint “address_street_id_fkey” on “address”
DETAIL:  Key (id)=(1) is still referenced from table “address”.
```

Вот такая она – ссылочная целостность.

PostgreSQL в лицах: Максим Валянский



Визитка LXF:
Сайт LOR (**http://www.linux.org.ru**) существенно отличается от других популярных форумов и новостных сайтов, хотя бы потому, что написан он не на PHP, а в качестве хранилища использует *PostgreSQL*. Об этом и пойдет разговор с создателем LOR **Максимом Валянским**.

Евгений Балдин: Стандартный вопрос, который в рамках цикла о PostgreSQL задается всем: «Почему *PostgreSQL*?» Ведь *MySQL* гораздо популярнее?

Максим Валянский: Изначально linux.org.ru разрабатывался как «хобби» проект, на котором хотелось попробовать интересные технологии. В октябре 1996 года *MySQL* была весьма примитивной СУБД, и работать с ней было неинтересно. Сейчас *PostgreSQL* прекрасно справляется с возрастающей нагрузкой и работает качественно и стабильно.

ЕМБ: На текущий момент используется *PostgreSQL 8.0*. Эта версия вышла примерно 2 года назад, а LOR существует уже свыше десяти лет. Что заставило перейти на «восьмерку»?

МВ: Мы стараемся почаще обновлять *PostgreSQL*. Как показывает опыт, каждая последующая версия *PostgreSQL* как минимум не хуже предыдущей, а зачастую оказывается, что некоторые наши запросы начинают выполняться быстрее из-за того, что разработчики учли подобные им при разработки новой версии. В целом *PostgreSQL* радует тем, что релизы качественно тестируются, и нам пока не приходилось «откатываться» на предыдущую версию из-за каких-то мешающих нам работать ошибок.

ЕМБ: Я понимаю, что «глас толпы» – это еще то указание к действию, но все равно хотелось бы спросить. На странице http://www.linux.org.ru/view-vote.jsp?vote=174 проводился опрос: «LOR FAQ — Нужен ли он в том виде, в котором существует сейчас?» С большим отрывом лидирует «Нет, лучше wiki на l.o.r». Если wiki будет, то на каком движке и как это будет соотноситься с *PostgreSQL*?

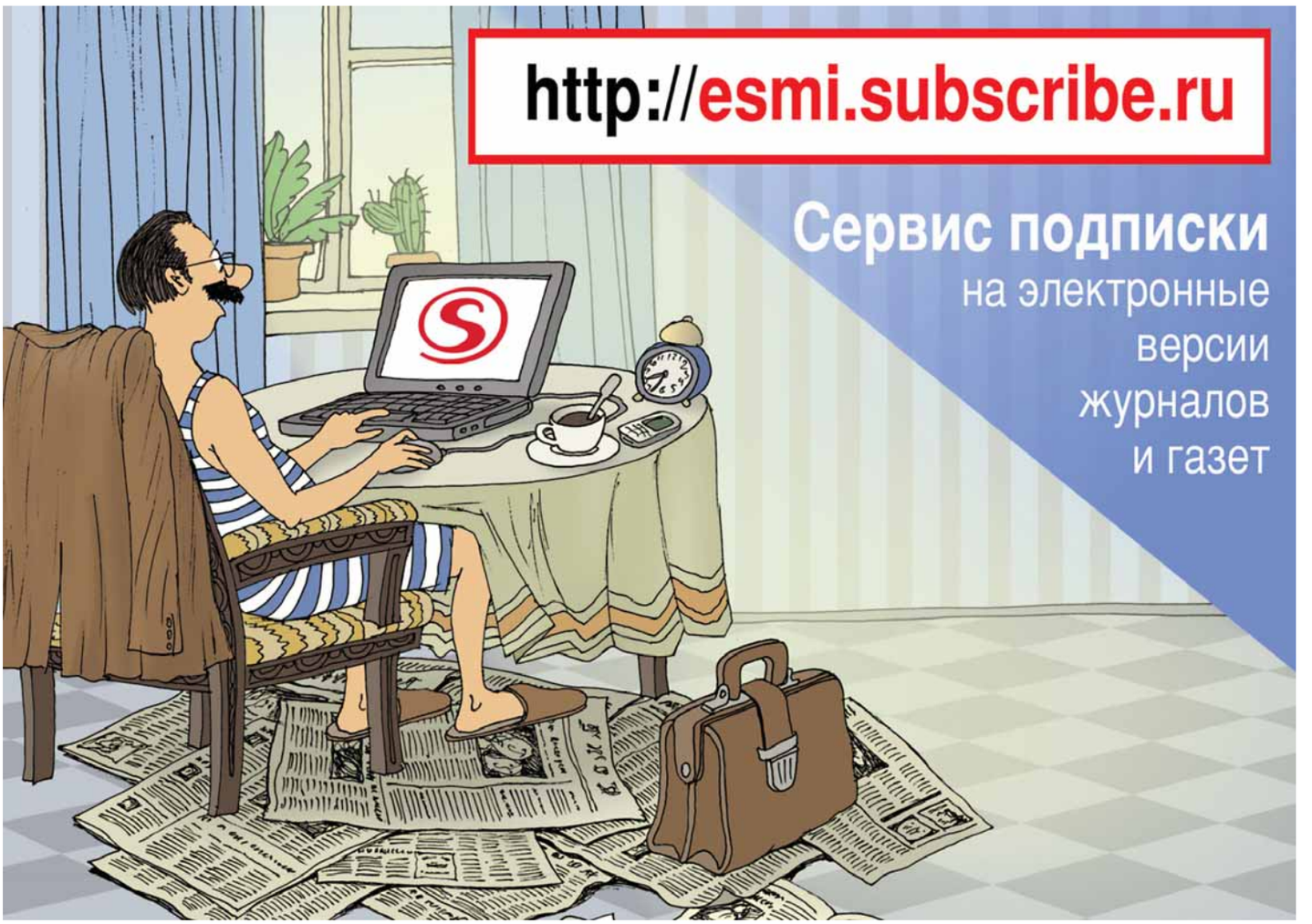
МВ: Мне бы хотелось, чтобы движок был написан в виде Java-сервлета и использовал *PostgreSQL*, для того, чтобы его можно было лучше интегрировать с основным сайтом, однако я пока не исследовал имеющиеся движки.

ЕМБ: Кстати, почему именно Java? Подавляющее число форумных, новостных и wiki-сайтов сделаны на PHP. Тоже потому, что было интересно?

МВ: Да. Изначально сайт был разработан на PHP, потом мне стало интересно изучить Java.

Mandriva FLASH
Mandriva Linux на 2Gb USB-накопителе

Подари Linux любимому человеку
Защити любимых
от вирусов и троянцев,
предложи им свободу выбора
и десктоп будущего
в маленьком, но умном
USB-накопителе.
Linux Center
www.linuxcenter.ru
Закажите
продукты Mandriva
в Линуксцентре!
www.linuxcenter.ru/mandriva



В «Системном администраторе» вы не прочтете о:

– котировках валют
– сплетнях
– погоде
– политике
– развлечениях

В вашем распоряжении:

– опыт лучших IT-специалистов
– новые идеи и полезные советы
– самые эффективные решения в области системного и сетевого администрирования

Подпишитесь сейчас!

Роспечать – 20780, 81655
Пресса России – 87836
Online-подписка – www.linuxcenter.ru

Время подписки ограничено!

Готовим

ЧАСТЬ 8 Эффектная презентация может привести к чему угодно – от докторской степени по орнитологии до головной боли в зале. **Евгений Балдин** расскажет, как добиться желаемого эффекта средствами LaTeX.

Существует три разновидности людей:
те, кто видит; те, кто видит, когда им показывают;
и те, кто не видит.
Леонардо да Винчи

Хочется показать свою крутизну? Подкупи слушателей. Хочется донести свою идею? Сделай нормальную презентацию.

При этом вовсе не нужен аляповатый фон, мультипликация при смене слайдов, но необходим разборчивый текст и картинки к месту. Вполне можно ограничься «прозрачками» и стандартным «оверхэдом». Если слайд требует от аудитории размышления, то на него следует не пожалеть как минимум пяти минут. В противном случае все ваши усилия будут напрасными Имея это в виду, давайте попробуем разобраться, что может LaTeX в сфере создания презентаций.

slides

Динозавр среди классов LaTeX, специализирующихся на презентациях. Идея очень проста: в качестве класса документа выбирается slides, в результате базовый размер шрифта автоматически увеличивается. Это позволяет прочитать стандартный текст на экране и избавиться от одного из смертных грехов докладчика – желания уместить слишком много информации на одной страничке. Опция класса landscape устанавливает альбомную ориентацию для страницы по умолчанию. Сами слайды создаются с помощью окружения slide. Вот, собственно, и всё.

› slides – это просто.

Идея обработки

Разрабатываются два независимых отбора событий по трековой системе с эффективностью ε_T и по калориметру с эффективностью ε_C, тогда

$$\varepsilon_T = \varepsilon_T^{(C)} \cdot \varepsilon_C + \varepsilon_T^{(\bar{C})} \cdot (1 - \varepsilon_C), \qquad (1)$$

где $\varepsilon_T^{(C)}$ — эффективность отбора T при условии, что отбор C сработал (она определяется экспериментально на данных J/ψ), а $\varepsilon_T^{(\bar{C})}$ — эффективность того же отбора при условии, что отбор C не сработал (Определяется на $J/\psi \to \pi^+\pi^-(X)$).

Аналогично

$$\varepsilon_C = \varepsilon_C^{(T)} \cdot \varepsilon_T + \varepsilon_C^{(\bar{T})} \cdot (1 - \varepsilon_T). \qquad (2)$$

1

```
\documentclass[a4paper,landscape]{slides}
 …
\begin{document}
\begin{slide}
 \begin{center}
  \Large Идея обработки
 \end{center}
 …
\end{slide}
\end{document}
```

Класс seminar похож на slides и лишь чуть-чуть более современен (1993 г.), но не в пример лучше документирован (файл **sem-user.pdf**) и кроме стандартного окружения slide имеет простейший набор команд для создания рамок.

В общем, если вам нужно по-быстрому сготовить презентацию из уже имеющегося текста, и все что вы хотите – это просто продемонстрировать идею, классы slide и seminar окажутся весьма кстати.

beamer

Время шло, компьютеры матерели, появились проекторы и захотелось чего-то разноцветного. Так появилось новое поколение презентационных классов.

С помощью пакета *beamer* в принципе можно создавать «прозрачки», как это делается посредством *slides*, но основное его предназначение – электронная презентация. Пакету чуть более трёх лет, но он очень активно развивается и на сегодня это, пожалуй, лучший выбор для создания презентаций в LaTeX. Его автор, Тиль Тантау [Till Tantau], оказался очень восприимчивым к предложениям со стороны сообщества. У *beamer* есть масса стандартных стилей, исчерпывающее описание на более чем двухстах страницах (**beameruserguide.pdf**) и официальный сайт: **http://sourceforge.net/projects/latex-beamer**.

beamer можно использовать как с *pdflatex*, так и со связкой *latex* + *dvips* + *ps2pdf*. При желании можно даже использовать *beamer* в связке с *LyX*. *TeX Live* включает в себя *beamer* по умолчанию. Для установки на системах Debian следует выполнить команду:

```
> sudo apt-get install latex-beamer
```

После этого в начале преамбулы можно выбрать класс beamer, примерно так:

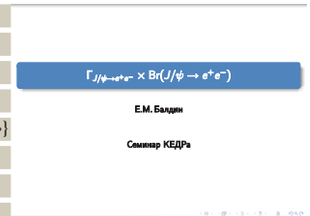
```
\documentclass[hyperref={unicode=true}]{beamer}
\usepackage[koi8-r]{inputenc}
```

По умолчанию, класс beamer загружает пакет *hyperref*. Если в документе планируется использовать этот пакет с установками, отличными от стандартных, то их следует передать в качестве необязательного параметра команды выбора класса. Если текст представлен в кодировке UTF-8, это также необходимо указать при загрузке *beamer*:

```
\documentclass[utf8]{beamer}
\usepackage[utf8]{inputenc}
```

Теперь можно выбрать тему для презентации и определить заголовок для титульного листа. Единицей представления для *beamer* является окружение frame:

```
% выбор темы
\usetheme{Madrid}
\useoutertheme{shadow}
\title{«Заголовок»}
\date{«Дата или место проведения»}
\author{«Автор»}
\begin{document}
% титульная страница
```



Месяц назад Мы познакомились с классами LaTeX и решили подробнее остановиться на презентациях.

презентации

Немного о PDF

За прошедшие годы Microsoft PowerPoint (PPT) стал де-факто стандартом для представления презентаций. Сегодня его потихоньку начинают теснить OpenDocument Presentation (ODP) и другие форматы на основе XML. LaTeX напрямую не поддерживает ни те, ни другие, зато позволяет легко сохранить созданные вами слайды в формате PDF. PDF – Portable Document Format – открытый платформенно-независимый формат для описания документов, созданный компанией Adobe Systems, Inc. ещё в 1993 году. В январе 2007 года началась процедура ратификации PDF как стандарта ISO. В 2006 году была опубликована версия стандарта под номером 1.7. Файл в формате PDF может представлять из себя комбинацию векторной графики, текста и растровых изображений (фотографий, снимков экрана и тому подобного). В стандарте PDF предусмотрена возможность создания гиперссылок, заполняемых форм и интерактивных вставок на JavaScript. Начиная с версии 1.6, декларируется возможность описания трехмерных интерактивных документов. Что бы это ни означало, звучит заманчиво, но к сожалению, использовать эти возможности пока рано.

С точки зрения формата для представления презентации, PDF удовлетворяет всем необходимым условиям, таким как:

- **Простота создания.** Это сила качественных открытых форматов – рано или поздно их начинают поддерживать все, кому не лень.
- **Переносимость.** Программу для просмотра PDF можно найти буквально везде.
- **Элементы интерактивности.** Документ может представляет из себя не только плоскую последовательность страниц.

Простота создания

Допустим, что PostScript-файл презентации уже был получен тем или иным способом. С помощью ghostscript, а точнее – скрипта ps2pdf (man ps2pdf) из него легко можно получить нормальный PDF:

```
> ps2pdf «файл.ps» «файл.pdf»
```

Получить PDF можно и напрямую из исходных текстов – с помощью программы *pdflatex*. Ее основное отличие от *LaTeX* как раз и состоит в том, что на выходе получается не DVI, а полноценный PDF. При использовании *pdflatex* следует учитывать, что графика должна быть представлена либо в виде PDF (вектор), либо PNG/JPEG (растр). *pdflatex* не умеет обрабатывать EPS-файлы, за исключением картинок, созданных при помощи *MetaPost*.

В PDF можно внедрять векторные шрифты Type1. Это позволяет отображать готовые документы, независимо от набора шрифтов, установленных в системе. Отображение на экране, особенно при низких разрешениях, зависит исключительно от качества внедрённых шрифтов. Парадокс качества: чем хуже разрешение, тем больший объём работы надо проделать с векторным шрифтом, чтобы он выглядел приемлемо. К счастью, в случае презентаций это не является проблемой, так для повышения читабельности на большом экране размер шрифта нужно значительно увеличить. Это эффективно увеличивает разрешение до сравнимого с разрешением лазерного принтера, для которого и оптимизированы наиболее популярные векторные шрифты Computer Modern (пакет cm-super). Для отображения на экране ни в коем случае не следует использовать растровые шрифты в формате Type3. Обязательно нужно установить шрифты cm-super, заведомо присутствующие в *TeX Live*.

Если вдруг по какой-то причине PDF потребуется преобразовать в PostScript, лучше всего воспользоваться утилитой *pdftops* из пакета *xpdf*:

```
> pdftops [-eps] «pdf-файл»
```

Если необходимо получить картинку в формате EPS, следует использовать ключ -eps.

Переносимость

Adobe Reader (ранее Acrobat Reader) и Ghostscript есть везде (если этой парочки где-то еще нет, то она легко может там появиться). Adobe Reader предоставляется всем желающим самой Adobe Systems. Как следствие, в смысле поддержки всех расширений формата PDF эта программа «впереди планеты всей». Поэтому презентацию, скорее всего, придётся показывать с помощью неё.

Одной из раздражающих особенностей Adobe Reader, мешающей использовать его при работе над документом, является отсутствие возможность перезагрузить изменённый файл. Эту проблему можно частично решить с помощью сторонних программ *pdfopen* и *pdfclose*, включенных в дистрибутив TeX Live:

```
> pdfclose --file «файл.pdf»
# обновляем «файл.pdf»
> pdfopen --file «файл.pdf»
```

Ghostscript и его программа просмотра не менее вездесущи. Ghostscript отображает PDF как обычный «плоский» документ, то есть об интерактивных эффектах можно забыть. Зато проблем с обновлением текста нет: нажал «.» (точку) и картинка обновилась.

Просмотрщик PDF-документов xpdf (http://www.foolabs.com/xpdf/) доступен только для систем с X Window. Начиная с *xpdf 3.02*, программа поддерживает структуру PDF вплоть до версии 1.7. *xpdf* используется как «движок» и для других программ просмотра, например, *KPDF*. Обновить документ можно с помощью клавиши R. Он очень удобен при просмотре в процессе подготовке документа.

Интерактивность

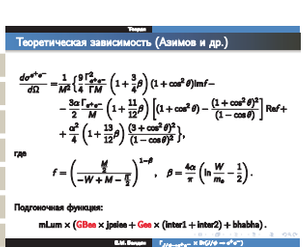
Зависит исключительно от стиля, который используется для подготовки PDF. Присутствует весь простейший джентльменский набор: гиперссылки, различные виды переходов со слайда на слайд и анимация. Есть и ограниченная возможность демонстрировать клипы и внедрять в презентацию звук.

```
\begin{frame}
  \titlepage
\end{frame}
```

Окружению frame можно передать необязательный параметр t, который «прижимает» текст к верхней части слайда.

Теперь можно приступать к самой презентации. Как и в обычных статьях, в *beamer* можно применять команды структурной разметки типа section. Эти команды должны использоваться за пределами окружения frame. Структурная разметка, в частности, полезна для организации быстрого доступа, например, через оглавление. Оглавление создаётся с помощью стандартной команды \tableofcontents. Ей можно передать необязательный параметр pausesections, чтобы оглавление разворачивалось не сразу, а по ходу дела.

```
%структурная разметка
\section{Теория}
\begin{frame}
  %заголовок слайда
  \frametitle{Теоретическая
    зависимость (Азимов и др.)}
  …
  \alert{GBee}…\alert{Gee}
  …
\end{frame}
```

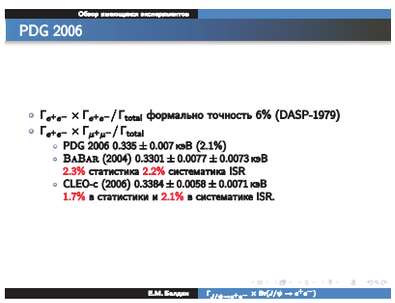


Для создания заголовка текущего слайда используется команда \frametitle. Команда \alert является аналогом \emph. По умолчанию выделенный сегмент просто отображается красным цветом, но при желании \alert всегда можно переопределить.

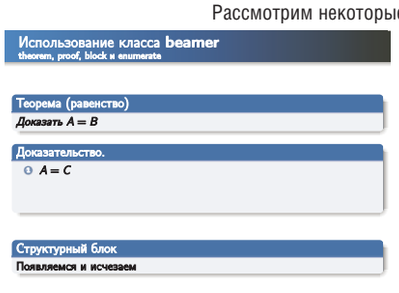
Оверлеи

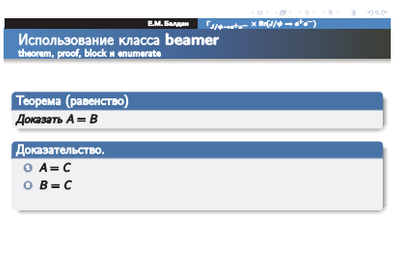
В процессе представления бывают очень полезны оверлеи – составные слайды, которые как бы накладываются друг на друга. Для создания простейшего оверлея используется команда \pause.

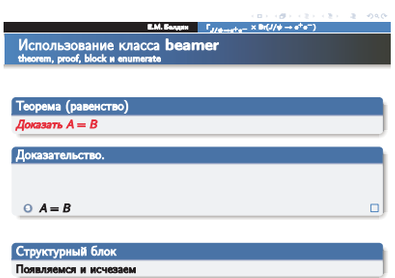
```
\begin{itemize}
  \item …
  \pause
  \item …
\end{itemize}
```

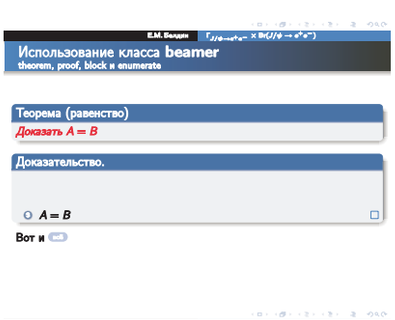


В *beamer* предусмотрена масса способов работы с оверлеями. Рассмотрим некоторые из них:










```
%создание своей теоремы
\newtheorem{rustheorem}{Теорема}

\begin{frame}
\frametitle{Использование
        класса \textbf{beamer}}
%подзаголовок
\framesubtitle{theorem, proof,
          block и enumerate}
%теорема
\begin{rustheorem}[равенство]
 \color<3-4>[rgb]{1,0,0}
          {Доказать \(A=B\)}
\end{rustheorem}
%доказательство
\begin{proof}
\begin{enumerate}
 \item<-2> \(A=C\)
 \item<2> \(B=C\)
 \item<3,4> \(A=B\)\qedhere
\end{enumerate}
\end{proof}
%последняя фраза
\uncover<4->{Вот и \beamerbutton{всё}}
%манипуляция с блоком
\begin{block}<1,3>{Структурный блок}
 Появляемся и исчезаем
\end{block}
\end{frame}
```

Для работы с оверлеями в *beamer* добавлен ещё один способ передачи параметров командам: < > – «меньше/больше». Таким образом команде передаётся список оверлеев, на которые она должна действовать. То есть, команда \color<3-4> раскрашивает текст в указанный цвет с 3-го по 4-й оверлей. Список можно передавать через запятую или как интервал. Записи: -3,5-9,12,17- означают, что команда действует для оверлеев из интервалов: от начала до 3-го, от 5-го до 9-го, для 12-го, от 17 и до конца.

Списки оверлеев могут принимать некоторые переопределенные команды. В их числе:

- \color{текст} – цвет текста.
- \item – определена внутри перечислений, к которым относятся окружения itemize и enumerate.
- Окружение theorem. Команда \newtheorem позволяет легко создавать свои теоремы.
- Окружение prof. Если есть теорема, то должно быть и доказательство. В конце доказательства традиционно добавляется квадратик – знак QED (quod erat demonstrandum – что и требовалось доказать). Команда \qedhere размещает QED в той же строке, где она указана, в противном случае для QED будет отведена своя собственная строка, что нежелательно.

Кроме этого, в классе beamer определены новые команды, воспринимающие список оверлеев, например:

- \alert{текст} – выделение текста.
- \only или \visible – добавление текста только для указанного списка оверлеев.
- \invisible – антипод \only.
- \uncover – тоже, что и \only, только под текст резервируется место даже на тех слайдах, где он отсутствует.
- \alt<список>{текст}{альтернативный текст} – для указанного списка оверлеев выводится «текст», для остальных – «альтернативный текст».
- Окружение block – именованный блок. Во многом аналогично окружению theorem.

Гиперссылки

Для создания гиперссылки необходимо установить в нужном месте метку или якорь. Это можно сделать с помощью команды \label. Собственно гиперссылка организуется с помощью команды \hyperlink:

```
\label{metka}
…
\hyperlink{metka}{«Гиперссылка»}
```

Вместо обычного текста можно использовать практически любую LaTeX-структуру, например, команду для создания «кнопки» \beamerbutton. Более общей командой для установки метки является \hypertarget:

```
\hypertarget<«номер оверлея»>{«метка»}{«текст»}
```

С её помощью можно указать не только структурную единицу, но и оверлей, на который следует сослаться.

Программный код

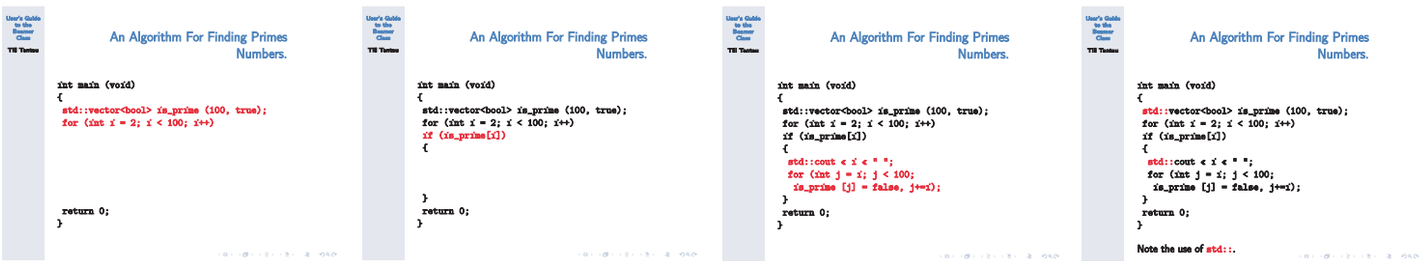
Для представления программного кода необходимо использовать окружения типа verbatim или lstlistings. Для того, чтобы код на слайде отобразился правильно, окружению frame необходимо передать опцию fragile. Оформление кода может выглядеть, например, так:

```
\begin{frame}[fragile]
%определяем более короткие команды
\newcommand{\un}{\uncover}
\newcommand{\al}{\alert}
\frametitle{An Algorithm For Finding Primes Numbers.}
\begin{semiverbatim}
\un<1->{\al<0>{int main (void)}}
\un<1->{\al<0>{\{}}
\un<1->{\al<1>{ \al<4>{std::}vector<bool> is_prime(100,true);}}
\un<1->{\al<1>{ for (int i = 2; i < 100; i++)}}
\un<2->{\al<2>{ if (is_prime[i])}}
\un<2->{\al<0>{ \{}}
\un<3->{\al<3>{  \al<4>{std::}cout << i << " ";}}
\un<3->{\al<3>{  for (int j = i; j < 100;}}
\un<3->{\al<3>{   is_prime [j] = false, j+=i);}}
\un<2->{\al<0>{ \}}}
\un<1->{\al<0>{ return 0;}}
\un<1->{\al<0>{\}}}
\end{semiverbatim}
\visible<4->{Note the use of \alert{\texttt{std::}}.}
\end{frame}
```

Выбор и настройка темы

Темы в *beamer* разбиваются на пять классов:

- **Именные темы** – концепция презентации. Для выбора темы используется команда \usetheme. Обычно создатель именной темы просто выбирает в ней соответствующие цветовую, шрифтовую и декоратив-



ные темы. По состоянию на начало 2007 года в beamer существовали следующие именные темы: AnnArbor, Antibes, Bergen, Berkeley, Berlin, Boadilla, CambridgeUS, Copenhagen, Darmstadt, Dresden, Frankfurt, Goettingen, Hannover, Ilmenau, JuanLesPins, Luebeck, Madrid, Malmoe, Marburg, Montpellier, PaloAlto, Pittsburgh, Rochester, Singapore, Szeged и Warsaw.

» **Цветовые темы** – палитра презентации. Для выбора темы используется команда \usecolortheme. Можно выбирать из следующего набора палитр: albatross, beaver, beetle, crane, dolphin, dove, fly, lily, orchid, rose, seagull, seahorse, sidebartab, structure, whale и wolverine.

» **Шрифтовые темы** – выбор подмножества шрифтов. Для выбора темы используется команда \usefonttheme. Существуют следующие шрифтовые темы: professionalfonts, serif, structurebold, structureitalicserif и structuresmallcapsserif.

» **Текстовые и структурные декорации** – темы, определяющие как выглядят перечисления, теоремы и выделения. Для выбора темы используется команда \useinnertheme. Доступны следующие варианты декораций: circles, inmargin, rectangles, rounded.

» **Внешние декорации** – темы, определяющие вид заголовков и обрамления слайда. Для выбора темы используется команда \useoutertheme. В вашем распоряжении – следующие типы обрамлений: infolines, miniframes, shadow, sidebar, smoothbars, smoothtree, split и tree.

Кроме этого, можно создать свою собственную тему и назвать её в честь родного города или страны. Подробности о том, как это сделать, следует искать в документации к пакету.

Ускорение компиляции

При подготовке презентации при выборе класса можно использовать опцию draft – это немного ускорит компиляцию. Можно также указать, какие именно слайды следует компилировать (это похоже на \includeonly):

```
\includeonlyframes{ex1,ex3}
\frame[label=ex1]
{Этот слайд будет включён при компиляции. }
\begin{frame}[label=ex2]
Аналогично ex2.
\end{frame}
\frame{А вот этого слайда не будет.}
```

Команда \againframe и метки позволяют повторно выводить уже имеющиеся слайды:

```
%ex1 будет выведен ещё раз
\againframe{ex1}
```

Печать слайдов

На самом деле, размер слайдов составляет всего 128x98 мм, то есть большие буквы получаются просто уменьшением размера листа бумаги. Для печати на листе формата A4 проще всего установить соответствующий параметр в Adobe Reader. Как вариант, чтобы всё гарантированно печаталось нормально, можно воспользоваться стилевым файлом **pdfpages** из пакета *pgf*:

```
\usepackage{pgfpages}
\pgfpagesuselayout{resize to}[a4paper,border shrink=5mm,landscape]
```

Здесь мы растягиваем слайд на страницу A4 в альбомной ориентации с 5-миллиметровым отступом от краёв. Если необходимо распечатать по два слайда на странице, можно поступить так:

```
\pgfpagesuselayout{2 on 1}[a4paper,border shrink=5mm]
```

Мультимедиа

Пакет *beamer* включает стилевой файл **multimedia**. Загрузив его, можно включить в презентацию клип или звуковое сопровождение, воспользовавшись командами \movie и \sound. К сожалению, пока эта возможность поддерживается только Adobe Reader в сборке для Windows и MacOS. Поддержка мультимедиа включена в стандарт PDF, поэтому её рано или поздно научится воспроизводить xpdf, даже если Adobe System «не почешется». Подробности об использовании этих команд можно посмотреть в пользовательской документации.

Наконец, в пакете *beamer* предусмотрена возможность создания анимации на основе созданных слайдов. Команда

```
\animate<«список оверлеев»>
```

позволяет автоматически проигрывать последовательность кадров. Для того чтобы эта функция заработала, необходимо раскрыть окно Adobe Reader на весь экран.

Правила хорошей презентации

Создание презентации – это очень тяжёлое занятие, так что не следует жалеть о потраченных на наведение блеска минутах. Готовя презентацию, не следует также забывать об эмпирически выведенных правилах:

» Один слайд требует не меньше одной минуты;

» Один слайд «со смыслом» требует не менее пяти минут;

» Времени всегда не хватает;

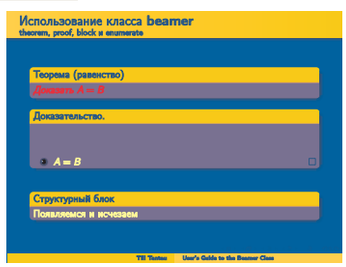
» Не следует «пихать» в презентацию больше слайдов, чем позволяет время. Перебор по времени только раздражает слушателей;

» Каждый слайд должен иметь свой заголовок (\frametitle);

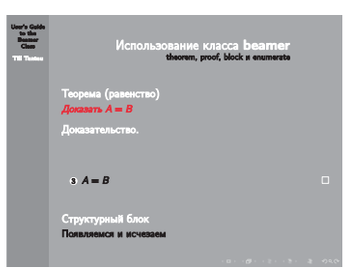
» В один слайд можно поместить около 20-40 слов, но заведомо не больше 80;

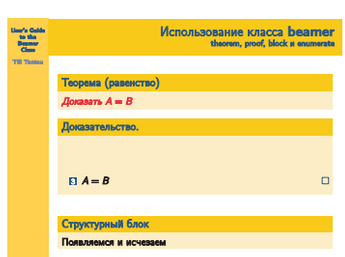
» Полезно использовать block, theorem, proof и example. Эти окружения структурируют текст и помогают выделять основные мысли;

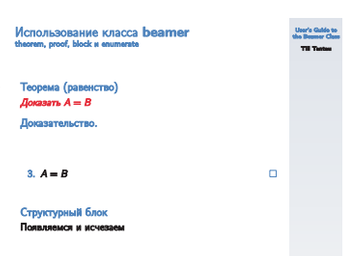
» Для разных аудиторий правила могут отличаться.

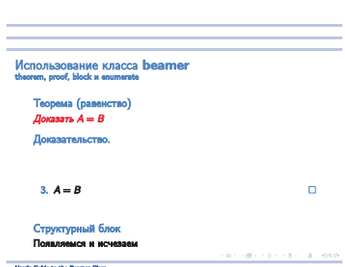


» **Представление программного кода (тема Hannover).**









» **Примеры разных тем *beamer* – малая часть от того, что есть.**

» **Через месяц** Вы познаете всю силу справочно-поискового аппарата издания

Создаем

ЧАСТЬ 4 До завершения нашей первой сцены осталось совсем чуть-чуть – нужно просто добавить качественную анимацию и освещение. **Андрей Прахов** научит ходить даже пингвина!

Вот мы с вами и добрались до заключительной части уроков *Blender* для начинающих. На этот раз нам предстоит покорить вершину 3D графики – анимацию персонажа. Недаром считается, что если ты сумеешь заставить естественно двигаться «живой» объект, то уже близок к ступеньке мастерства. Помимо работы со скелетом, мы поговорим о правильном освещении и добавим некоторые атрибуты для завершенности нашего детища. Опять-таки, советую запастись не только терпением, но и старыми выпусками LXF с предыдущими уроками.

Для работы нам понадобится либо специально подготовленный файл с диска LXF (**less4_begin.blend**), либо вы можете открыть собственный проект, над которым работали в последний раз, и немного модернизировать его.

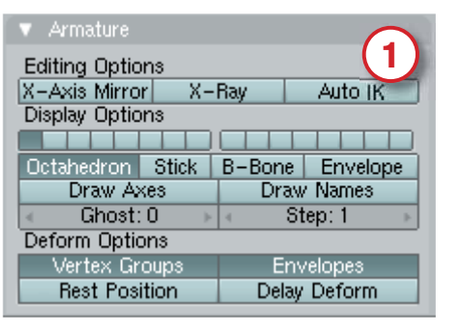
В первую очередь необходимо расчистить поле деятельности. Перенесите все лишние объекты, кроме пингвина, в любой удобный для вас слой. Саму модельку расположите в первом слое. На последнем уроке для удобства размещения детали объекта были сгруппированы. На данном этапе нам это не нужно, так что выделите модель и разгруппируйте ее (Object -> Group -> Remove from All Groups). Отметьте туловище, нажмите F9 и в закладке Modifiers нажмите кнопку Apply. Это означает, что действие, задаваемое модификатором, окончательно применяется к объекту и изменению уже не подлежит. Осталось только объединить модель в общий mesh-объект (Ctrl+J).

Анимация персонажа начинается с создания для него скелета. По аналогии с живыми организмами, арматура объекта состоит из совокупности соединенных между собой костей, призванных обеспечить «естественную» анимацию. Затем с помощью специальных инструментов, созданный скелет «вживляется» в модель объекта, настраиваются места сгибов, добавляются управляющие рычаги.

Для создания арматуры воспользуемся пунктом меню Add -> Armature, доступном по клавише Space (пробел). Перед этим перейдите в окно Front View и в свободной зоне справа от пингвина установите курсор на уровне шеи объекта. При этом создается начальная косточка, которая по умолчанию является предком всего скелета. Этот объект состоит из основания (самой кости) и шарниров. Именно шарниры обеспечивают связь и взаимодействие с другими костями. Для создания следующего звена нажмите E. Таким способом можно выстроить необходимую вам конструкцию.

Как и любой другой объект в *Blender*, Armature подчиняется общим правилам манипуляции. Вы можете перемещать, масштабировать любую из частей арматуры, вращать, экструдировать и т.д. Попробуйте уцепиться за какой-либо из шарниров и потянуть. Заметили, что вся конструкция, так или иначе, приходит в движение?

Будучи объектом, арматура имеет и свои функции, которые вызываются по клавише F9. Обратите внимание на следующие опции – они пригодятся нам в дальнейшем (см. **Рис. 1**):



Editing Options (опции редактирования):

- X-Axis Mirror Как известно, скелет любого «живого» объекта симметричен. Было бы накладно создавать арматуру, одинаковую для правой и левой половины. Эта функция позволяет зеркально отражать создаваемую конструкцию.
- X-Ray Режим «рентгена». Кости внутри объекта становятся видимыми, что очень удобно при анимировании.
- Auto IK Программа пытается просчитать инверсную кинематику арматуры в автоматическом режиме. Это не всегда приводит к положительным результатам, но для обкатки конструкции весьма полезно.

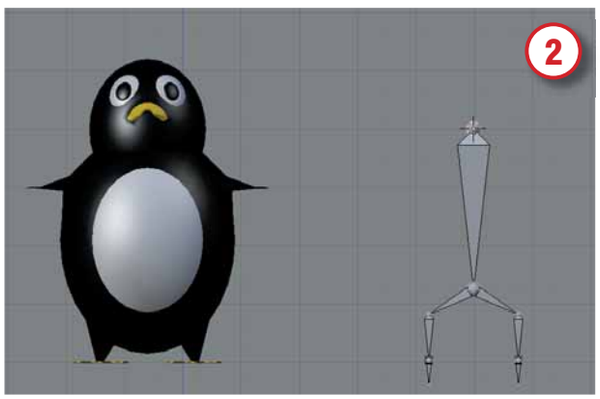
Display Options (опции отображения):

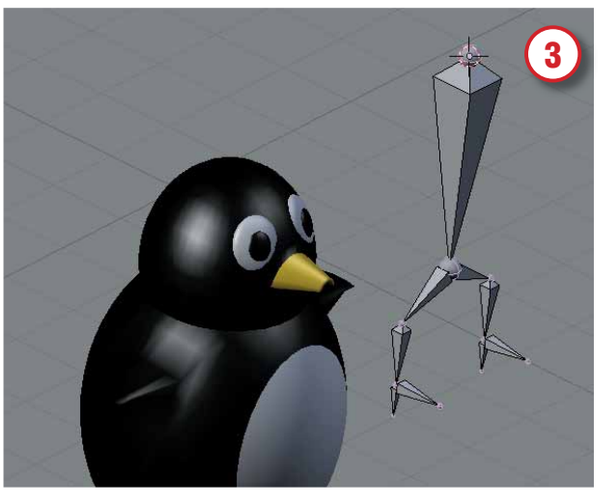
Это группа параметров, влияющих на отображение объекта на экране.

- Octahedron, Stick, B-Bone, Envelope – форма костей.
- Draw Axes – визуализация осей подобъектов.
- Draw Names. Каждое звено в арматуре имеет свое уникальное имя, генерируемое программой или пользователем. Эта функция отображает на экране имена, присвоенные элементам конструкции.

Месяц назад Мы познакомились с ключевой анимацией и текстурированием.

скелет





Сейчас настало время создать нашу первую арматуру. Она должна будет не только позволять модели двигаться, но и имитировать характерное покачивание живых пингвинов при ходьбе. На данном уроке мы ограничимся только этим, но заложенная база позволит вам впоследствии воплотить самые безумные телодвижения.

Создайте конструкцию в соответствии с **рисунками 2** и **3**. Не забудьте активировать кнопку X-axis Mirror. Для создания парных элементов используйте сочетание клавиш Shift+E. Следите, чтобы кости располагались строго по осям, без отклонений. Проще всего это сделать, если при движении зажать среднюю кнопку мыши.

На протяжении всех уроков я упорно говорил о важности разумного именования объектов в сцене. Это отнюдь не моя прихоть. Дело в том, что в некоторых случаях, для нормальной работы программы, необходимо пользоваться определенными правилами создания имен. И вот этот момент настал! Для правильной автоматической настройки кинематики арматуры это просто жизненно необходимо, иначе программа будет вести себя совсем не так, как мы задумывали.

При выделении любой косточки становится активной закладка Armature Bones. Поменять текущее имя вы сможете в строке BO:. Но перед этим активируйте режим отображения Draw Names. Сделайте все это в соответствии с **рисунком 4** (кроме того, обратите внимание на врезку «Правила именования звеньев арматуры»).

Правила именования звеньев арматуры

Программа чувствительна к регистру символов, поэтому при вводе названий придерживайтесь следующих требований:

1 В имени могут присутствовать все буквы и цифры, но не символ «точка».

2 Буквы «R» и «L», отделенные от имени точкой обозначают, соответственно, звенья правой и левой стороны. Они обязательно должны быть заглавными.

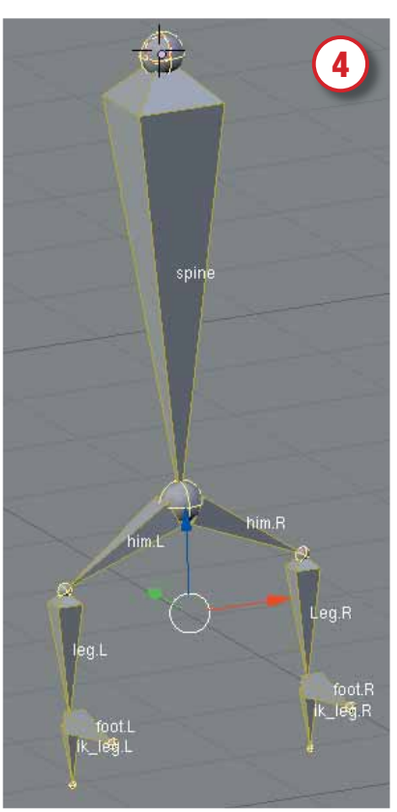
3 Используйте характерные имена для облегчения восприятия. Это уже просто желательно и считается хорошим тоном.

Примеры правильных имен: «**name.L**» или «**name_name.L**».

Анимация персонажа

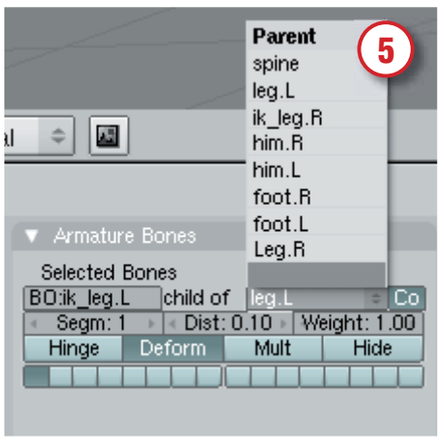
Для работы с арматурой во время настройки анимации существует специальный режим Pose Mode. Активировать его можно простым выбором из списка режимов объекта, либо нажатием Ctrl+Tab.

Давайте немного поиграем со скелетом. Попробуйте подвигать разные части арматуры. Заметили, что при движении звена перемещается лишь то, что находится ниже в иерархии? Это происходит из-за того, что не настроена обратная кинематика. Теперь включите кнопку Auto IK и сравните с прежним результатом. Налицо какой-то прогресс: скажем, если взяться за spine и подвигать, то наш скелет примется весело танцевать. Но это и все, что предлагает нам автомат. Пришло время ручной настройки.

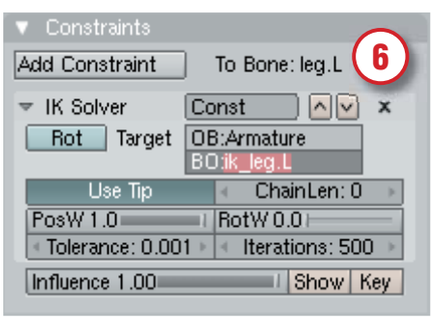


Обратите внимание на звенья ik_leg.L и ik_leg.R. Эти кости, кажущиеся совсем чужими и ненужными наростами на скелете, являются своеобразными рычагами. Управление арматурой не непосредственно несущими звеньями, а специально созданными для этого рычагами считается хорошим тоном. Это удобно и с точки зрения видимости происходящего.

Теперь необходимо немного перестроить структуру иерархии в нашей арматуре. Войдите в режим редактирования и выделите звено ik_leg.L. В закладке «Armature Bones» существует запись BO: ik_leg.L child of leg.L которая гласит, что данная кость является производной от звена leg.L. Из выпадающего списка справа можно выбрать любую другую косточку – и она станет родительской. Но мы выберем «пустоту», тем самым отсоединив звено от арматуры (**рис. 5**).

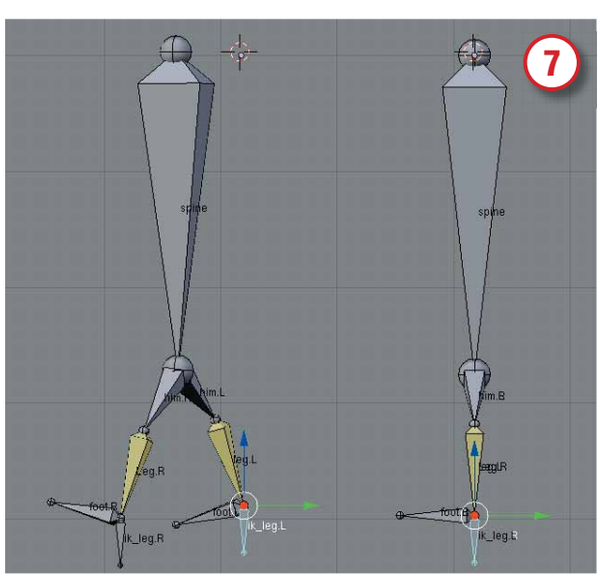


Проделайте ту же самую операцию и для ik_leg.R. Для настройки обратной кинематики перейдите в режим Pose Mode и выделите leg.L. Теперь в закладке Constrains будет доступна кнопка Add Constraint, используемая для указания контроллера. Выберите из списка IK Solver. Обратите внимание, что выделенный объект изменил свой цвет с синего на оранжевый. При правильной настройке контроллера он поменяет свой цвет на более яркий. Введите в поле OB: значение «Armature», а в BO: – ik_leg.L. Теперь становится понятным мое требование давать звучные имена объектам. К сожалению, в этой версии редактор не поддерживает выбор из выпадающего меню, все приходится править вручную (**Рис. 6**).

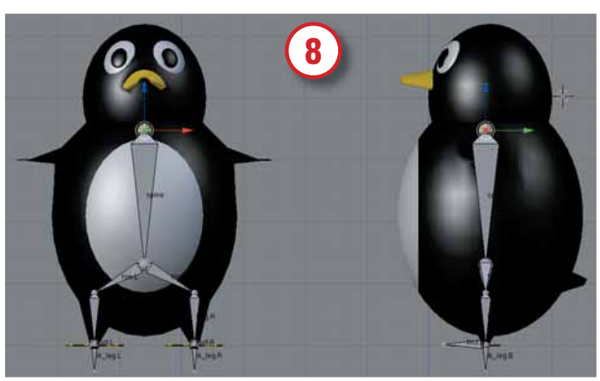


Проделайте тоже самое и с костью leg.R, только в качестве рычага выберите ik_leg.R.

Попробуйте передвинуть ногу арматуры, ухватившись за рычаг Ik-контроллера (Pose Mode, Rear View). Если ничего не вышло и скелет начало быстро вращаться вокруг оси или был сильно искажен, проверьте, отжата ли кнопка Auto IK, в противном случае откатитесь назад и настройте все заново. Вполне вероятно, что движение арматуры покажется вам неестественным. Для более точной подстройки, попробуйте поменять параметр ChainLen для каждой ноги (leg.L, leg.R). Не забудьте, что при движении туловище (spine) также необходимо перемещать вслед за конечностью (**Рис. 7**).



Теперь нам предстоит натянуть на каркас модель и настроить места сгибов. Для этого верните арматуру в начальное положение, отменив все ваши изыскания по движению, и совместите ее с пингвином (чтобы не нарушить расположение модели, двигайте арматуру). При помещении скелета в объект он перестает быть виден. Вот тут то и пригодится режим X-Ray! Разместите арматуру так, как показано на **рисунке 8**. Для подгонки смело пользуйтесь масштабированием и редактированием отдельных звеньев, но не нарушайте при этом симметрию.

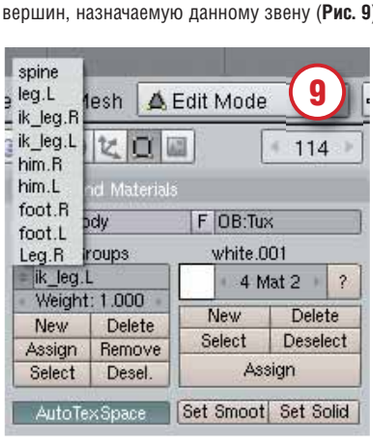


Чтобы объединить модель и арматуру, нужно установить последнюю в качестве родительской в иерархии, а затем привязать нужные группы вершин к костям. Для этого выделите пингвина, затем скелет (именно в этой последовательности!) и нажмите Ctrl+P -> Armature. Появится меню из трех пунктов:

1. «Don't create groups» – не создавать группы. Программа просто установит иерархическую связь. Если вспомнить тягу редактора к ручному вводу, то этот путь самый неудобный. Необходимо создавать группы вершин по количеству костей, причем их имена должны совпадать.
2. «Name groups» – создание групп по существующим костям. Все группы будут названы также, как и их предки. Наиболее удобный способ, сочетающий в себе простоту и гибкость.
3. «Create from close bones» – создать по ближайшим костям. Полностью автоматический режим. Чреват неверными определениями привязок вершин.

Мы воспользуемся вторым пунктом, чтобы внимательно рассмотреть весь этот процесс. После выбора, выделите пингвина, нажмите Tab для перехода в режим редактирования и включите прорисовку wireframe (Z).

Задача состоит в следующем: для каждой группы необходимо указать область вершин, назначаемую данному звену (**Рис. 9**).

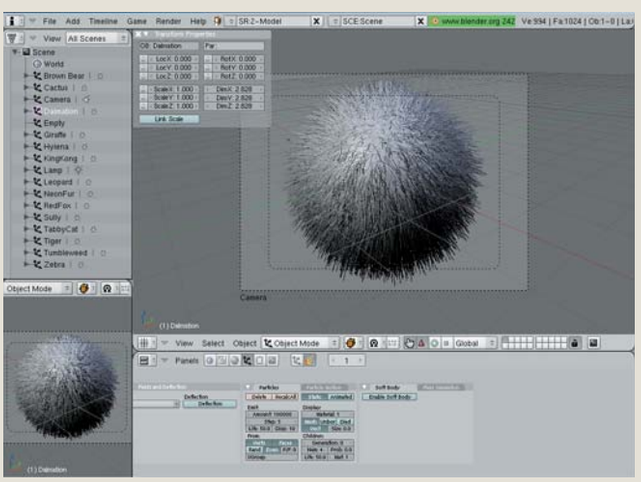


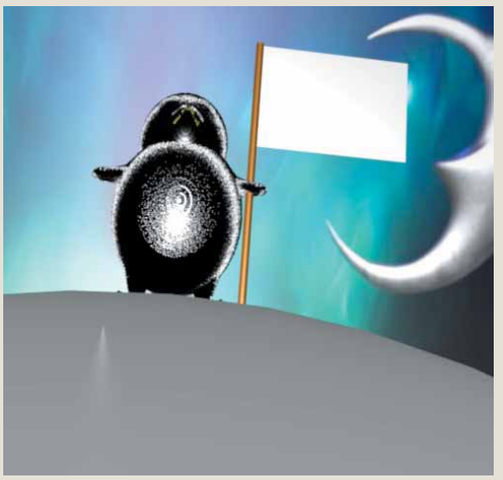
Комментарий Александра Супрунова

Одной из самых замечательных особенностей Blender, существенно повышающих его мощь, является возможность расширения за счет сторонних модулей и библиотек. Благодаря им, создание морской ракушки, дерева или шестереночного механизма часов превращается в щелчок мышью и несколько секунд ожидания.

Мы пока не будем заселять земной шар, вращающийся под ногами Тукса. Но вот сделать помохнатее самого пингвина можно и даже нужно! Да, я говорю про мех.

Для достижения этой цели, воспользуемся расширением *BrianHFurLibVol1_13July2006*. Это обычный blend-файл, содержащий библиотеку, основанную на эффекте многих частиц, воспроизводящих волосяной покров жирафа, далматинца, леопарда, зебры и даже... Кинг Конга.





Для добавления «меховой» библиотеки достаточно нажать Shift+F1 и выбрать файл **BrianHFurLibVol1_13July2006.blend** с диска LXF.

Щелкните по нему, чтобы войти внутрь и перейдите в папку Object. Укажите тип меха: tiger. На одном из слоев должен появится мохнатый шар.

Находясь в режиме Object Mode, выделите пингвина вместе с мохнатым шаром (удерживайте клавишу Shift). Теперь нажмите Ctrl+C и в появившемся меню выберите Particle Settings – Тукс покроется густым мехом. Его длину и прочие параметры можно подрегулировать во вкладке Particles, Для этого нужно нажать F7 и активировать на панели кнопку Physics buttons. В частности, параметр Life отвечает за длину меха, а Amount – за его густоту. Не бойтесь экспериментировать!

Работа с группами вершин была подробно описана ранее – при создании модели (LXF89). Сделайте следующие привязки:

- foot.L и foot.R – стопы объекта.
- leg.L и leg.R – подъем от стопы к туловищу.
- spine – все, кроме стоп.

Вот и все. Для проверки, выделите арматуру и перейдите в режим позы. Для проверки «походки», попробуйте подергать за рычаги в окне Rear View. При этом должны двигаться только ноги со ступнями. Теперь перейдите в Front View и покачайте spine справа налево. Здесь, наоборот, происходит качание всего туловища без стоп. Если при движении неправильно работают вершины, нужно всего лишь удалить или добавить оные в нужные группы.

Осталось вернуть все спрятанные объекты на место и настроить анимацию. Как вы уже догадались, для этого мы воспользуемся возможностями ключевой анимации (LXF90). Перейдите в режим позы и, совмещая движение земного шара с ногами, создайте несколько ключей. Не забудьте, что наш пингвин должен не двигаться, а перебирать ногами. Настроив этот вид ходьбы, перейдите в окно Front View и добавьте характерное качание при ходьбе, причем на те же самые ключи, что и созданные ранее.

Если вам удалось настроить анимацию и пингвин зашагал вперевалку, то, наверное, вы заметили, что флаг все так же упорно остается на месте и за движениями модели не следует. Это дело поправимое. Выделите древко знамени, затем косточку spine (арматура должна быть в режиме Pose) и нажмите Ctrl+P для создания иерархии. Выберите из предложенного меню пункт Bone. Все! Теперь наша анимация настроена окончательно.

Искусство освещать события

Освещение в трехмерных сценах – это целая наука, которая заслуживает отдельной немаленькой статьи. Но, тем не менее, мы и сейчас, особо не вникая в теорию освещения, сможем значительно улучшить восприятие нашей сцены. Посмотрите вокруг себя, сколько источников света вы сможете насчитать? Солнце, экран монитора, всевозможные отражения – даже крохотный экран мобильника и то вносит свою лепту. А теперь подумайте о переносе всего этого в компьютер. Пожалуй, нереально, да и мощностей наших с вами машин не хватит. Но не все так тоскливо, и 3D-дизайнеры уже давно наловчились «обманывать» народ. Речь идет о так называемом окружающем свете, который равномерно пытается осветить все объекты в сцене. Попробуйте отрендерить кадр из сцены (F12). Тускло и безжизненно. Теперь нажмите F5, затем крайнюю справа кнопку World buttons (иконка в виде земного шара). Вы попадете в панель настройки «окружающего мира».

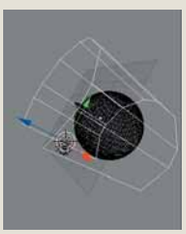
Ambient Occlussion – вот то, что нам нужно. Нажав на эту кнопку, вы включите окружающий свет. Выполните рендеринг и сравните картинку с предыдущей. Налицо значительное улучшение общего вида... и столь же значительно увеличилось время обработки.

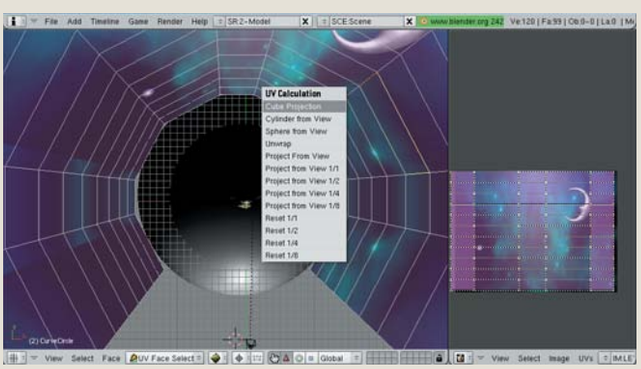
Так как у нас в сцене явно присутствует космическая составляющая, то неплохо было бы добавить звездное небо. Это можно сделать «в »

Комментарий Александра Супрунова

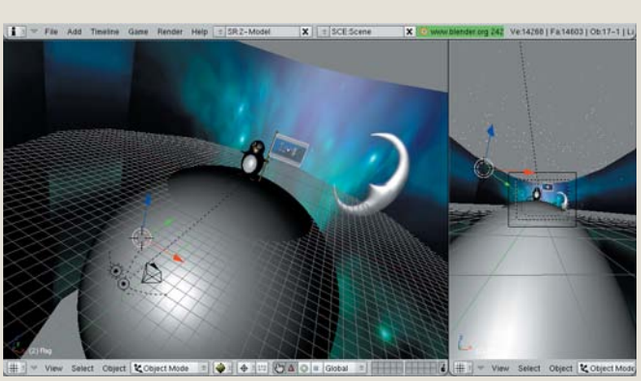
Для создания звездного неба на этом уроке мы пошли простейшим путем и настроили параметры Mist/Stars.... Альтернативный подход аналогичен используемому в компьютерных играх – окружающее пространство представляется кубом с натянутыми текстурами звезд, туманностями и т.д.

1 Но поскольку мы не пишем динамичную стрелялку, а занимаемся моделированием, время отрисовки сцены не столь критично, и использовать куб нет особой нужды. Возьмем для разнообразия трубу, Tube (Add->Mesh->Tube).
Увеличим размер трубы (S), чтобы внутри нее поместилась вся сцена с пингвином. Чтобы получить большую свободу действий, можно удалить часть трубы, расположенную перед камерой, выделив точки (B).





2 Напомню также, что если какие-то объекты мешают моделированию, не обязательно переносить их в другой слой – достаточно просто спрятать, нажав клавишу H. Чтобы увидеть спрятанные объекты снова, нажмите Alt+H.
Добавив в сцену трубу, сделайте ее гладкой, открыв нужную вкладку (F9) и нажав кнопку Set Smooth. Теперь давайте наложим на нее текстуру.
Для этого разделим экран на два окна. В левом перейдите в режим UV Face Select. В правом выберите UV/Image Editor.
В окне UV/Image Editor вызовите Image->Open и откройте картинку с текстурой звездного неба. Колесиком мыши уменьшите проекцию, чтобы она была видна в окне целиком.
Теперь переместите курсор мыши в левое окно и нажмите U, чтобы сделать развертку объекта для наложения текстуры.



3 Появится большое меню, предлагающее выбрать метод развертки – очевидно, что для нашего случая больше всего подходит Cylinder from View. Чтобы увидеть наложенную на объект текстуру, нажмите Alt+Z.

4 А теперь обратите внимание на окно UV/Image Editor – в нем вы увидите развертку. Можете масштабировать ее, вытягивать и т.д., наблюдая за изменениями, происходящими на текстуре, натянутой на объект, в реальном времени.
Чтобы увидеть наложенные таким образом текстуры, необходимо сделать небольшие изменения в режиме Object Mode. Нажмите F5, а затем, в меню Material нажмите кнопку TexFace, а во вкладке Map Input – UV.

5 Теперь можно произвести отрисовку сцены обычным образом.

лоб», внедрив на задний план подходящую картинку, либо воспользовавшись некоторыми заготовками программы.

Находящаяся рядом закладка Mist/Stars... предоставляет подобную возможность. Нажмите кнопку Stars, затем кнопку Blend в окне Preview и обработайте картинку.

Вот и все. Мы завершили работу над нашей первой сценой и завершили этим цикл статей, посвященных начальным ступенькам в мир 3D. Время не стоит на месте, уже сейчас вышла следующая версия *Blender*'а, оснащенная новыми, еще более мощными и удобными инструментами. Угнаться за прогрессом невозможно, но, тем не менее, основа успеха нами, я думаю, была заложена. Удачи! LXF

Ваше мнение

Вам понравилась серия уроков по Blender и вы хотели бы освоить и более сложные приемы? Сообщите нам об этом, отправив письмо по адресу letters@linuxformat.ru

Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

В этом месяце мы отвечаем на вопросы по:

1. Определении диска SATA
2. DVD-слайд-шоу
3. Символических ссылках
4. Кодировании звука
5. Настройке GUI
6. Системных шрифтах
7. Беспроводной сети
8. Настройке экрана
9. Инсталляции
10. Сканировании
11. Освобождении порта
11. DansGuardian
13. MonoDevelop
- Правах доступа
- Восстановлению разделов

1 Военная хитрость

Пожалуйста, помогите! Меня зовут Джек. У меня проблема.

Моя материнская плата – P4S800D, на чипсете SIS655FX, и у меня два диска: на одном, IDE, стоит OpenSUSE 10.2, на другом, SATA – Windows. Инсталлятор SUSE 10.2 определяет только диск IDE. Как настроить и смонтировать SATA в OpenSUSE 10.2? Я нашел драйвер на официальном сайте SIS, но получил ошибку *make*, так как она не смогла найти scsi_request.h. Проблема ядра?

Джек [Jack]

Привет, Джек! SATA пока что вызывает проблемы у многих. По нашему опыту, простейшее решение – переключить диски в режим совместимости через BIOS, закончить инсталляцию и попробовать вернуть все обратно. Многие дистрибутивы затрудняются «встать» на обычные SATA-диски, но после установки прекрасно работают – особенно после закачки новейших заплаток. Убедитесь также, что у Вас не установлен программный RAID, с ним тоже бывают проблемы. В крайнем случае, попробуйте добавить **insmod=ide-generic** во время инсталляции. Удачи! **ПХ**

2 Видео слайд-шоу

Я собрался сделать видео-CD из моих фотографий. Для начала хочу просто поместить фото на видео-CD, чтобы можно было просматривать их на обычном DVD-плейере или телевизоре. Потом добавлю звук.

Похоже, что инструментов для этого хватает, но мне не удалось найти четкого описания этого процесса. Например, с *FFmpeg* удалось сделать из моих фотографий кадры длительностью около 0.4 с. Мне надо, чтобы каждый кадр длился около 3 секунд (например), но я не смог найти способа изменить длительность. Convert выглядит многообещающе, но не дружит с *mpeg2encode*.

Превратите чудесные воспоминания а-ля Анита Экберг в слайд-шоу с *DVD-Slideshow*.

У меня Ubuntu Dapper. Заранее благодарен за любые советы.

Daudi

Использовать DVD было бы лучше: и больше фото поместится на один диск, и качество будет заметно выше. Процесс в основном одинаков для DVD и видео-CD, но все инструменты ориентированы на DVD, поэтому для видео-CD потребуется перенастройка.

Простейший способ поместить слайд-шоу на диск – это воспользоваться слайд-шоу *DigiKam* или *KPhotoAlbum* (оба используют один и тот же модуль расширения) для создания DVD-слайд-шоу из фотоальбома или отдельных фотографий. Возможности их ограничены: можно лишь менять задержку между кадрами и время затухания – да и то единообразно для всех фотографий.

Если хочется большего, то *DVD-Slideshow* (домашняя страница **http://dvd-slideshow.sourceforge.net**) – лучший выбор. Это набор скриптов для генерации DVD из картинок и звука. Главный скрипт, *dvd-slideshow*, использует список изображений и эффектов в текстовом формате, для формирования DVD VOB-файла. Создайте входной файл *DVD-Slideshow* с помощью *dir2slideshow*: его можно отправить прямо на *dvd-slideshow* или отредактировать для изменения длительности кадров и эффектов. Затем организуйте слайд-шоу и добавьте музыку с помощью dvd-slideshow. Просмотреть VOB-файл перед записью на диск можно через MPlayer.

Наконец, *dvd-menu* может создать – угадайте, что? – DVD-меню для одного или более слайд-шоу; также есть возможность вызова **dvdauthor** для изготовления ISO-образа, готового к записи на диск. Допустим, директория, из которой нужно сделать слайд-шоу, называется **pics**. Тогда понадобятся команды:

```
mkdir slideshow
```

Наши эксперты

Мы найдем эксперта в любом вопросе! Вы получите ответ на все: от проблем с установкой или модемом до сетевого администрирования; главное – спросить!

Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.

Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.

Пол Хадсон
Пол – местный супер-программист. Он управится со всеми вашими проблемами по части web и баз данных.

Ник Вейч
В свободное от исчеркивания текстов красными чернилами время Ник возится с Linux-графикой и 3D-приложениями; он у нас отвечает за простые вопросы!

Владимир Иванов
работает с Linux уже более 10 лет, днем – создавая комплексные решения для крупных компаний, а вечером – модерируя сайт linux.org.ru и отвечая на вопросы о системном администрировании и информационной безопасности.

Валентин Синицын
В свободное от работы время редактор нашего журнала разрабатывает *KNetworkManager* и другие открытые приложения. Он с радостью поможет вам в вопросах использования Linux на рабочем столе.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите на м по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

```
dir2slideshow -o slideshow -t 5 -c 1 –n myslideshow pics
# отредактируйте myslideshow.txt для настройки длительности или эффектов
dvd-slideshow -a somemusic.ogg myslideshow.txt
dvd-menu -t “My slide show” -f myslideshow.xml –iso
```

Мы получим слайд-шоу с длительностью кадра 5 с и затуханием 1 с, записанное в ISO-образ, готовый для записи на DVD. Можно изготовить DVD с единственным слайд-шоу, воспроизводимым непосредственно, без меню. По умолчанию формат выхода – NTSC; для формата PAL DVD понадобится добавить к каждой команде **-p** или поместить

```
pal=1
```

в ~/.dvd-slideshowrc.

Если нужно создать DVD-совместимый MPEG, воспользуйтесь *FFmpeg* для перекодирования созданного Вами в *dvd-slideshow* VOB-файла, вот так:

```
ffmpeg -target pal-vcd -i dvdslide.vob vcdslide.mpg
```

НБ

3 Потерянная ссылка

Мне необходимо создать ссылку с одной директории на другую, чтобы при обращении программы к директории *X* ей предъявлялась директория *Y*. Я пробовал *ln* с различными параметрами, но она всякий раз создает ссылку внутри целевой директории.

Причина, по которой мне это нужно – обновление с *OpenOffice.org 2.0* до *OOo 2.1*, в результате чего создалась новая директория **/opt/openoffice.org2.1**. Когда я щелкаю мышью на текстовом документе или таблице в KDE, программа пробует войти в **/opt/openoffice.org2.0**, которой больше нет. Если я делаю **cd** в **/opt** и набираю

```
ln -s openoffice.org2.0 openoffice.org2.1
```

создается символическая ссылка *OpenOffice.org 2.0* внутри директории 2.1. Я уже все перепробовал – не работает, и точка!

OnlyTheTony

В Вашей работе с *ln* – две проблемы. Во-первых, синтаксис: правильно будет ‘ln -s источник назначение’. Я тоже попадал в эту ловушку: когда я переходил с ОС, использующей обратный порядок, некоторое время приходилось думать дважды. Аргументы должны идти в том же порядке, что для *cp* или *mv*: мне так легче было запомнить.

Другая проблема – если в качестве целевой указать существующую директорию, *ln* предполагает, что ссылку Вы хотите создать внутри нее. Это перекликается с *cp* и *mv*, они тоже копируют или перемещают файл внутрь целевой директории. Удалите целевую директорию, и *ln* создаст нужную ссылку:

```
ln -s openoffice.org2.1 /opt/openoffice.org2.0
```

Заметьте, что в символьных ссылках источник задается относительно назначения, поэтому даже если эта команда исполняется не в директории **/opt** – а значит, директория или файл **OpenOffice.org2.1** не существует – *ln* все равно сработает.

А еще можно зайти в Центр управления KDE, раздел файловых ассоциаций, и указать верный путь вызова программ *ooffice2*. **НБ**

4 Количество и качество

У меня 64-битный Ubuntu 6.10 на AMD64, я занимаюсь кодированием звука. Я затеял небольшой тест, чтобы посмотреть, что эффективнее: последовательное кодирование четырех директорий с FLAC-файлами (четыре файла одного размера в каждой директории) в OGG или параллельное. Я написал скрипты на *Bash*, чтобы измерить производительность. Первый скрипт работал около девяти минут (чуть больше двух минут на директорию), но второй тоже занял девять минут, хотя в каждой директории был девятиминутный объем кодирования.

Я уверен, что с какого-то момента параллельное выполнение задач становится медленнее, чем последовательное. Просматривая вывод *top*, я заметил, что при параллельной обработке четыре экземпляра flac были запущены одновременно, и каждый занимал около 20% процессорного времени. При последовательной работе единственный процесс *flac* использует намного больше возможностей CPU.

Проводились ли какие-либо тесты на эту тему? При работе с тысячами файлов приходится экономить время.

Пол Хок [Paul Hoch]

Распараллеливание задач требует накладных расходов, обусловленных переключением между ними и управлением памятью, но при небольшом количестве задач это несущественно. Если Вы попробуете запустить одновременно 20 или 30 процессов, то заметите значительное замедление, особенно если будет задействован раздел подкачки [swap].

Кодирование файлов с одного жесткого диска на другой сильно нагружает процессор и память, мало затрагивая сами диски – спецы называют эти задачи ‘compute-bound’ или ‘CPU-bound’ (то есть, CPU-зависимыми). Напротив, извлечение данных с CD или DVD сильно зависит от скорости передачи, а требования к процессору невысоки – такие задачи называют ‘IO-bound’ (зависимыми от ввода-вывода). Поэтому выполнение двух однотипных задач ‘CPU-bound’ или ‘IO-bound’ одновременно почти не дает преимущества по сравнению с последовательной работой, а вот одновременная работа двух процессов разной зависимости значительно повышает производительность.

Если Вы кодируете аудио с оптики или другого источника с малой скоростью передачи, можно значительно ускорить процесс распараллеливанием, примерно так:

```
Rip track 1
Encode track 1 in the background
Rip track 2
```

Многие CD-ripперы/кодеры так и делают, например, мои любимые *Grip* (**www.nostatic.org/grip**) для графической среды или Abcde (**www.hispalinux.es/~data/abcde.php**) для консоли. Если аудиофайлы уже на жестком диске, не запускайте много процессов одновременно, но держите как минимум два – один процесс с необходимостью будет прерываться.

Единственный полезный тест – тот, что близок к реальным условиям, то есть выполнение Вашей собственной работы с хронометражем, как Вы и поступили.

Краткая справка о...

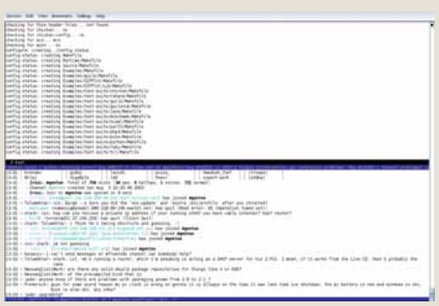
Screen

Интеллектуальное управление терминалами в Linux.

Освоившись с командной строкой Linux, вы будете пользоваться терминалами все чаще, иногда несколькими одновременно. Вкладочный *X*-терминал – неплохой способ слежения за несколькими терминальными сессиями, но от него не много толку при удаленном доступе и виртуальной консоли. *Screen* – оконный менеджер для терминалов, позволяющий запускать несколько сессий на одном терминале; но не только.

Наберите в терминале **screen**. С виду это похоже на очистку экрана, но на самом деле вы окажетесь в другом терминале. Наберите **ls -l** для просмотра текущей директории, затем нажмите Ctrl+A и C – все исчезнет. Перейдите в другую директорию, снова запустите ls и нажмите Ctrl+A N – и вы опять в прежнем окружении. Ctrl+A и C создают новый терминал, оставляя прежний в работе, Ctrl+A и N переключают между ними. Теперь нажмите Ctrl+A и D для выхода из *Screen* и возврата в оболочку, затем откройте другой *X*-терминал и наберите **screen -r** для возврата в прежнюю сессию *Screen*.

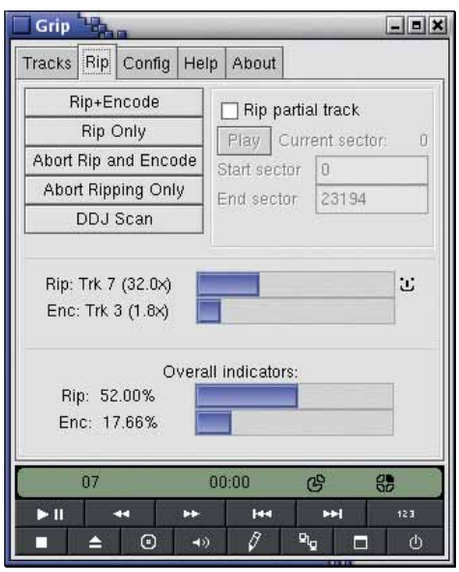
Screen не ограничивается локальными подключениями – он хорошо работает через SSH. Например, можно подключиться к домашнему компьютеру с работы, запустить *Screen*, загрузить новую версию некоторой программы, начать компиляцию и покинуть *Screen*. Позже можно снова войти в сессию *Screen* с помощью **screen -r** (там же, на работе, или вернувшись домой) и узнать, как идут дела. Даже когда SSH-подключение закрыто, программы, запущенные в *Screen*, продолжают выполняться.



В верхней части экрана можно компилировать программу, а заодно болтать по *IRC* «этажом ниже».

Есть масса других возможностей; все они подробно описаны в man. Можно разделить окно с Ctrl+A и S; перейти в новую область с Ctrl+A и Tab; воспользоваться Ctrl+A и N (или C) для перехода в новый терминал *Screen*. Можно разделять окно до тех пор, пока будет свободное место, наблюдая выполнение команды во время взаимодействия с интерактивной программой в другой части окна.

» Не забывайте, что работа по кодированию проводится в фоне, поэтому, если у Вас большой объем или работа очень срочная – на тесты может уйти больше времени, чем можно сэкономить на повышении производительности компьютера. Вы уже установили, что распараллеливание малого числа процессов не дает результата. Увеличение их количества делу не поможет – разве что у вас несколько многоядерных процессоров. **НБ**



› ***Grip*** **использует ресурсы машины по максимуму, параллельно выполняя CPU- и IO-зависимые процессы.**

5 Где взять GUI

У меня стабильная Linux-система, являющаяся моим настольным ПК и обслуживающая малую домашнюю/офисную LAN. У меня есть несколько свободных разделов на жестком диске, где я пробую новые дистрибутивы. Из любопытства я поставил Fedora Core 6 (LXF88). Труднее всего в этих экспериментах было заставить работать беспроводную карту PCI. На ней стоит бесславный чипсет Broadcom BCM4318, который с Linux не очень-то дружит.

Следуя подсказкам и советам, для активации карты я сделал следующие шаги. Во-первых, инсталлировал драйверы с помощью *NdisWrapper*. Во-вторых, отключил драйвер Fedora BCM43xx. В-третьих, по инструкции SourceForge, отредактировал два сетевых файла [**modprobe.conf** и **icfg-eth0**]. Благодаря этому мой eth0-интерфейс работает подобно wlan0 в других дистрибутивах. Карта запускается из командной строки вот так:

```
/etc/init.d/network restart
```

Для завершения работы я вызвал NetworkManager из системного меню KDE. Хотел через диалоговое окно настройки сети нанести последние штрихи, но оно оказалось пустым. Никакого NIC-интерфейса нет, но при этом вся система действует безупречно. Я просто купаюсь в техническом и мультимедийном великолепии широкополосной сети. Как можно графически управлять тем, что уже сделано в темных глубинах системы из командной строки?

Джим МакФарлан [Jim Macfarlane]

О Хоть Вы и обращались с Вашей картой несколько нестандартно, она работает – браво! А Вы настроили псевдоним *NdisWrapper*, дав от имени root следующие команды?

```
ndiswrapper -ma
echo “alias wlan0 ndiswrapper” >> /etc/ modprobe.conf
```

А главное, создали ли Вы после всего этого новый сетевой интерфейс для устройства с помощью инструмента Fedora *system-config-network*? Если все это сделано, а *Network Manager* все еще не работает, попробуйте запустить его при загрузке, вот так (тоже от имени root):

```
chkconfig NetworkManager on
chkconfig NetworkManagerDispatcher on
```

Network Manager – совершенно новый инструмент, к тому же он постоянно развивается. Ваши проблемы могут быть решены в Fedora 7, ожидаемой в апреле. **ПХ**

6 Хочу большой!

В Я только что установил Fedora Core 6. Как увеличить системный шрифт?

Саид Фарах [Said Farah]

О О, это простой вопрос, я такие люблю! Системный шрифт в Fedora настраивается в меню System > Administration (Система > Администрирование), пункт Fonts (Шрифты). Когда появится окно Fonts Preferences (Параметры шрифта), щелкните на кнопке Details (Подробнее) в правом нижнем углу, затем найдите разрешение экрана в верхнем левом углу нового окна. Увеличение этого числа делает шрифт больше, а вместе с ним увеличивается размер кнопок, окон, меню и др. Не забудьте записать первоначальное разрешение – вдруг к нему придется вернуться. **ПХ** »

Часто задаваемые вопросы...

Rsync и Unison

Синхронизация двух директорий – движение не одностороннее!

» Что такое *rsync*?
Это способ синхронизации содержимого двух директорий, причем обе будут полностью идентичны. cp копирует все подряд, а *rsync* – только те файлы, которые различаются. Если изменились большие файлы, копируются только измененные части.

» Какой синтаксис у этой команды?

```
rsync --archive --delete /path/to/source/
/path/to/dest/
```

точно скопирует первую директорию во вторую. Параметр **--delete** удаляет файлы, которых нет в первой директории, а **--archive** копирует также права доступа и временные отметки. Замыкающие слэши важны: они указывают на то, что синхронизируется содержимое директорий. Без слэшей программа просто скопирует одну директорию в другую.

» Можно ли синхронизировать директории разных компьютеров? У меня локальная копия web-сайта, и не хотелось бы копировать все подряд, когда изменится только пара файлов.
Если у вас есть SSH-доступ к web-серверу, можно сделать так:

```
rsync --archive --delete /my/local/site
server:/path/to/site/
```

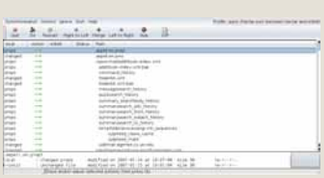
Если один из путей или оба содержат hostname компьютера, *rsync* использует для передачи удаленную оболочку. По умолчанию это SSH, но можно заменить ее любой другой – даже RSH, если вы не против, чтобы ваши файлы во время передачи читал кто ни попадя!

» А как синхронизировать директории между Linux и Windows-компьютерами?
rsync работает и под Windows, но параметр **--archive** на разделах FAT действовать не будет, ведь в FAT не сохраняются права доступа. Подробности можно узнать в man-странице программы.

» Что если файлы изменились на обоих компьютерах, и данные тоже нужно сохранить на обоих?
Для этого понадобится специальная программа, например, Unison (www.cis.upenn.edu/~bcpierce/unison). Для синхронизации *Unison* пользуется той же rsync, а для передачи – SSH, но она предназначена для двусторонней синхронизации. Если на двух компьютерах изменились некоторые файлы, каждый компьютер получит новейшую версию.

» Умно придумано; а если я отредактирую один и тот же файл на обоих компьютерах?
Программа не умеет читать мысли и решать, какая версия вам дороже, она просто сообщит, что обе копии редактировались, и спросит, какую из них сохранить. *Unison* отслеживает все изменения, и заметит, что файл был изменен на обоих компьютерах.

» Придется учить тысячу параметров командной строки?
Придется. А можно воспользоваться GUI, и просто указывать да щелкать мышью. GUI работает только на том компьютере, который инициирует передачу, поэтому можете использовать его для синхронизации со своим web-сервером, не забудьте только установить там *Unison*.



› **Синхронизируйте ноутбук и настольный компьютер через Unison.**

Вопрос-победитель (русская версия)

Сергей Васильевич получает подарочный сертификат на 1000 рублей от интернет-магазина LinuxCenter.Ru! Просим победителя выйти на связь с редакцией: info@linuxformat.ru

Семь раз примерь – один отрежь

Я пытался поставить Linux на ноутбук Bliss 4020. К моменту установки на жестком диске имелось три раздела: два в формате NTFS с различной информацией, и один неотформатированный, в начале диска. Попытки проинсталлировать некоторые варианты Linux закончились неудачей – 1,8 ГБ оказалось явно недостаточно для полноценной ОС. Столкнувшись с нехваткой места, я начал искать инструменты для решения этой проблемы и нашел Paragon Partition Manager 7.0.1274.

Я решил увеличить свободное место путем урезания одного из разделов NTFS. Процесс уменьшения раздела прошел безболезненно, а вот процесс перемещения раздела, к моему большому сожалению, закончился зависанием.

Теперь у меня четыре раздела. Первый немного увеличился. Второй стал меньше, но главное, в нем все перепутано, не говоря уже о том, что файлы не читаются. Некоторые программы заявляют, что они из 30 ГБ доросли до пары десятков ТБ.

В этом и состоит вопрос. Подскажите способ восстановления раздела, лучше конечно, "непроприетарный". Разумеется, на освободившееся место я ничего ставить не стал. Пользуюсь *Knoppix 3.7 RE*.

Сергей Васильевич

О Ситуация, конечно, непростая, но довольно распространенная. Бывает, что при выполнении низкоуровневых операций с диском, пользователь или программа ошибается. Так или иначе, доступ к данным может быть потерян.

Восстановление удаленных файлов в общем-то стало процедурой, привычной для многих пользователей – существует масса вспомогательного ПО для различных операционных систем, что заметно упрощает задачу. Однако, работа с разделами жесткого диска все еще остается уделом немногочисленных хакеров и требует более глубокого понимания сути проблемы, технологий и форматов данных, вовлеченных в процесс.

Что бы понять, как восстановить утраченные данные, рассмотрим процедуру «передвижения» раздела по диску.

В процессе «передвижения» производится поблочное копирование данных, начиная с той части раздела, которая находится «ближе» к пустой области. На приведенном рисунке раздел перемещается «вправо», поэтому копирование происходит «справа налево». Если бы раздел перемещался влево, копирование бы началось с начального, «левого» блока данных. Выбор направления и очередности копирования важен для обеспечения целостности данных. Если бы копирование производилось в ином порядке, перемещаемые данные могли бы переписать содержимое раздела до его копирования.

После того, как раздел будет целиком перемещен, программа вносит изменения в таблицу разделов жесткого диска. Она расположена в нулевом секторе и имеет следующие поля:

1. По адресу **0x0000** находится код начального загрузчика;
2. По адресу **0x01be** находятся 4 записи таблицы разделов жесткого диска (то, что под Linux видно, как **/dev/hdx1**, **/dev/hdx2**, **/dev/hdx3**, **/dev/hdx4**), каждая длиной в 16 байт;
3. По адресу **0x01fe** находится двухбайтовая сигнатура MBR (0x55 0xaa).

Если в системе используются расширенные разделы, то в начале каждого расширенного раздела помещается дополнительная таблица разделов, имеющая следующий формат:

1. По адресу **0x01be** находится запись о первом разделе размером 16 байт;
2. По адресу **0x01ce** находится запись о следующей расширенной таблице разделов, длиной в 16 байт;

Каждая запись таблицы разделов по смещению **0x0001** содержит трехбайтовый адрес начала раздела в формате **CYL/HEAD/SECT** (цилиндр/поверхность/сектор).

К сожалению, не было указано, перемещение какого раздела выполнялось: стандартного или расширенного. Если перемещался основной раздел, то все операции будут осуществляться только с главной записью таблицы разделов. Если же перемещался раздел, расположенный внутри расширенной области (логический диск в терминах Windows), то были затронуты и расширенные таблицы разделов.

Перед выполнением работ по восстановлению рекомендуется сделать резервные копии. Конечно, лучше скопировать посекторно весь жесткий диск, но если вы не можете позволить себе такой роскоши, сохраните хотя бы секторы с таблицами разделов.

Сделать это можно командой **dd**.

Загрузившись с компакт-диска Knoppix, сохраните основную таблицу разделов:

```
dd if=/dev/had of=/path/to/mbr/back.up bs=512
count=1
```

Для сохранения резервной копии расширенной таблицы разделов, необходимо передать команде **dd** параметр **skip**, указывающий ее смещение. Как было сказано выше, положение расширенной таблицы разделов можно выяснить в главной таблице разделов.

Значение параметра **skip** равняется LBA-адресу начала раздела. Что бы выяснить LBA-адрес, зная адрес в формате цилиндр/поверхность/сектор, воспользуйтесь следующей формулой:

```
LBA = ( ( CYL * HPC + HEAD ) * SPT ) + SECT – 1
```

где CYL – номер цилиндра, HPC – количество поверхностей в одном цилиндре, HEAD – номер поверхности, SPT – количество секторов на дорожку, SECT – номер сектора.

После того, как резервная копия будет создана, можно приступать к попытке восстановления данных.

На имеющемся в Вашем распоряжении диске с *Knoppix* есть утилита *testdisk*. Ее мы и будем использовать для восстановления таблицы разделов. Запустив *testdisk*, выберете создание нового файла журнала, затем – ваш диск и тип таблицы разделов (Intel).

В появившемся меню выбирайте пункт Analyze и Search – это позволит найти заголовки разделов. Затем сохраните новую таблицу разделов на жесткий диск, выбрав пункт меню write.

Если зависание *Paragon Partition Manager* произошло на этапе изменения таблицы разделов жесткого диска, то после выполнения этой процедуры, все ваши диски будут доступны. Если же на момент зависания, утилита выполняла копирование данных раздела, предстоит дополнительная работа.

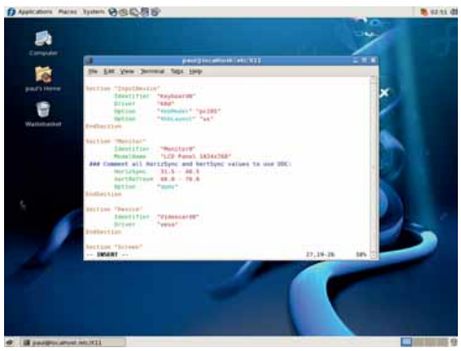
Как видно из рисунка, в процессе «сдвига» раздела образуется «дырка». Нашей задачей является «приклеить хвост раздела к голове» и восстановить данные. Дабы избежать случайного разрушения, рекомендую снова сделать резервную копию. Скопируйте весь раздел, включая «голову», «дырку» и «хвост» в отдельный файл. Далее мы будем использовать его для работы.

Обратите внимание, что при копировании блоков раздела утилитой перемещения образуются дублирующие блоки данных. В нашем случае, если зависание произошло в процессе записи, последний записываемый блок мог быть сохранен лишь частично. Исходя из этого, необходимо искать дубль блока, записанного предпоследним. К сожалению, компания Paragon Software не публикует данных о размере блока, используемого при работе, однако, начав с блока, равного одному сектору (512 байт), можно впоследствии уточнить результат, сравнив соседние блоки.

Для поиска можно использовать любой Hex-редактор или попробовать написать скрипт на Вашем любимом языке.

Обнаружив «начало конца раздела», вам останется лишь «приклеить» его к голове, используя все ту же команду **dd**, на сей раз – с параметрами **skip** и **seek**.

Желаю удачи! Но впредь – делайте резервные копии! **ВИ**



Изменяя разрешение экрана на рабочем столе Gnome, Вы увеличиваете размер шрифта.

7 FullMAC – это как?

Недавно установил Fedora Core 6 в двойную загрузку на мой HP Pavilion t3065 (Intel Pentium 4 3.4 ГГц с 1 ГБ ОЗУ). Все было прекрасно, пока я не попробовал соединиться со своим беспроводным G-модемом Belkin (802.11g – model F5D7632uk ver 1000). Долго чесал в затылке и копал в Интернете – и смекнул, что нужен беспроводной драйвер. Допросив сетевой контроллер, я определил следующие параметры чипсета:

- Intersil Corporation ISL3890 [Prism GT /Prism Duette] / ISL3886 [Prism Javelin /Prism Xbow] (rev 01).
- Subsystem: Accton Technology Corporation WN4201B.
- Flags: bus master, medium devsel, latency 64, IRQ169.
- Memory at cfffc000 (32 bit, non-prefetchable) Size 8k.
- Capabilities: (dc) Power management version 1.

Просмотрел множество разных Интернет-ресурсов на тему подключений с этим чипсетом (включая www.prism54.org), и растерялся. Какой драйвер установить – FullMAC или Islsm? Каждый драйвер пригоден для одного или другого варианта ISL, но не для обоих! Возникает вопрос: не все ли равно, какой брать?

Допустим, мне удалось связать Linux с моим модемом. Поддерживает ли Fedora или другой дистрибутив шифрование WPA-PSK, или пока нет ничего лучше 128-битного шифрования? Как можно использовать WPA-PSK на моем компьютере?

Джонатан Пис [Jonathan Peace]

Несколько лет назад Prism выпустила новую версию чипсета, перекладывавшего часть работы на основной процессор (другими словами, это был облегченный, удешевленный чипсет вроде используемых в win-модемах). Он стал известен как SoftMAC-дизайн, и нарушал совместимость с драйверами Prism54, пока не появился *Islsm*. Этот драйвер работает как с SoftMAC, так и с прежними чипсетами FullMAC. Драйвер FullMAC лучше работает с устройствами FullMAC, зато SoftMAC не признает вообще. К сожалению, трудно сказать, что именно есть у Вас: ISL3890 работает с драйвером FullMAC, а для ISL3886 необходим *Islsm*.

FullMAC включен в стандартное ядро Fedora Core 6. Вам нужно только установить прошивку (firmware), ее можно загрузить с http://prism54.org/fullmac.html и протестировать, открыв терминал и набрав

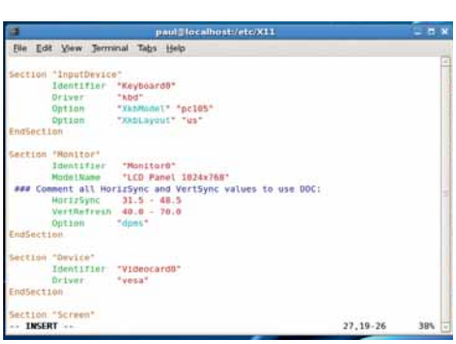
```
su
```

(в ответ на запрос, введите пароль администратора)

```
modprobe prism54
lsmod | grep prism54
```

Если вывод последней команды будет непустым, драйвер есть в наличии и загружен – попробуйте соединиться с модемом. Но сначала нужно отключить шифрование (WEP и WPA) – настройте соединение, а потом шифруйте (без работающего подключения и шифровать-то нечего). Если драйвер Prism54 не справится – попробуйте Islsm. Для этого тоже понадобится прошивка, но уже другая: ее можно взять на http://prism54.org/newdrivers.html. Подробные инструкции включены в пакет.

Шифрование WPA-PSK в Linux реализовано в форме *wpa_supplicant* (http://hostap.epitest.fi/wpa_supplicant). На Fedora Core пакеты для этого есть – нужно установить **wpa_supplicant и wpa_supplicant-gui**. Необходим только первый, второй нужен лишь для графического управления конфигурацией, чтения, редактирования и сохранения файлов настройки. **HB**



Обычно переключения на драйвер VESA для X достаточно для надежной работы двумерного рабочего стола.

Вопрос-победитель (английская версия)

Ограничение прав доступа

Можно ли ограничить число процессов, запускаемых одним пользователем в своей оболочке? У меня Ubuntu Dapper.

RD, из форумов LXF

Это делается двумя немного разными способами, в зависимости от того, использует ли Ваша система PAM (Pluggable Authentication Modules, Подключаемые модули авторизации). Ubuntu использует PAM по умолчанию, поэтому пределы устанавливаются в **/etc/security/limits.conf**. Чтобы ограничить Фреда десятью процессами, добавьте строчку вроде этой:

```
fred hard nproc 10
```

Для систем, в которых нет PAM, лимит можно установить в **/etc/limits**, подобным способом:

```
fred U10
```

В любом случае можно подставить вместо имени * (звездочку), чтобы ограничить всех, кроме администратора.

Лимиты, установленные в этих файлах, действительны для каждого входа в систему (а не для пользователя «в целом»), но не забывайте о том, что вход в графический рабочий стол сам по себе требует запуска нескольких процессов. Терминал, открытый из этого рабочего стола, по умолчанию не будет считаться отдельным входом, поэтому ограничивайте процессы умеренно. Чтобы узнать, сколько процессов запускает пользователь при стандартной загрузке, наберите

```
sudo ps -u fred | wc –l
```

Пример PAM включает жесткий лимит, а вообще PAM устанавливает два типа лимитов: жесткий и мягкий. Жесткий неоспорим, менять его разрешено только администратору – но пользователь может расширить лимит между мягким и жестким ограничениями командой *ulimit* (считайте мягкий лимит стандартным ограничением, а жесткий – абсолютным максимумом). Оба лимита можно уравнять знаком – во втором поле строки **/etc/security/limits.conf**. Ограничить можно не только число процессов: можно ограничить использование ресурсов памяти или процессора. См. man для **limits.conf** и *ulimit*, там можно найти (намного) более подробные сведения. **HB**

8 Nvidia не видит

У меня AMD64 3000+ CPU, с 1 ГБ ОЗУ, видеокарта Nvidia GeForce PCI-express и жесткий диск 320 ГБ SATA. Во время установки Fedora Core 6 все было хорошо, пока не дошло до «starting udev (OK)» – тут экран погас. Жесткий диск вроде бы работал, но экран был пуст; затем вышло сообщение: «Mode not supported».

Сперва я подумал, что виновата видеокарта, но при инсталляции Elive 0.5 все заработало. Попробовал снять видеокарту и пользоваться встроенной картой VIA – ничего; попробовал другой экран – ничего. Попробовал загружаться со всеми возможными параметрами, доступными для экрана в Fedora – ничего не работало. Пробовал принудительно установить разрешение экрана (linux resolution=1024x768) и запускал *linux noprobe*.

Среди сообщений об ошибках, которые можно отнести к данной проблеме, были следующие: «PCI: BIOS Bug: MCFG area at e0000000 is not E820-reserved» и «PCI:Not using MMCONFIG». Уж не знаю, могут ли они помочь, раз все это не мешает работать Elive. Не могли бы вы помочь мне с установкой Fedora?

Daryl

Похоже на то, что Fedora пробует активировать встроенный драйвер Nvidia, но он не в ладах с разрешением экрана. Быстро устранить проблему поможет переключение на драйвер VESA, который должен надежно работать с любой графической картой. Откройте **/etc/X11/xorg.conf** как администратор и найдите следующую строку:

```
Driver "nv"
```

Замените **nv** на **vesa** и перезагрузитесь. По крайней мере, получите рабочую систему.

А вот теперь, если VESA не годится для Вашей повседневной работы, если Вы хотите попробовать »

AIGLX или сыграть в 3D-игру, лучше будет установить официальный драйвер Nvidia с www.nvidia.com. Он намного стабильнее встроенного в Fedora, и должен устранить Вашу проблему. **НБ**

9 Что делать с DeLi Linux

Приятно было найти DeLi Linux на диске LXF86, ведь у меня есть 486 PC для пробы. Прожиг CD и запись загрузочной дискеты прошли удачно, и инсталляция продолжалась до тех пор, пока не появилась панель с вопросом: «Where is delibase.tgz?» (*Где находится delibase.tgz?*). Среди вопросов на панели был и такой: «I can scan for CD-Rom drives. Should I try to do so?» (*Можно просканировать CD-приводы. Сделать это?*).

Щелкнув Yes, получаю другую панель с предложением «Enter the device which contains the DeLi Linux Base Package delibase.tgz» (*Укажите устройство, на котором содержится основной пакет DeLi Linux, delibase.tgz*). Что бы я потом ни вводила, следовало сообщение: «ERROR! Failed to mount the source device. Exiting …» (*ОШИБКА! Невозможно смонтировать устройство. Выход…)* – и все. Скопировала этот файл на C:\ (одно из предлагаемых расположений на панели «Where is… и т.д.» – но когда пробую ввести **C:**, получаю лишь **C:#**.

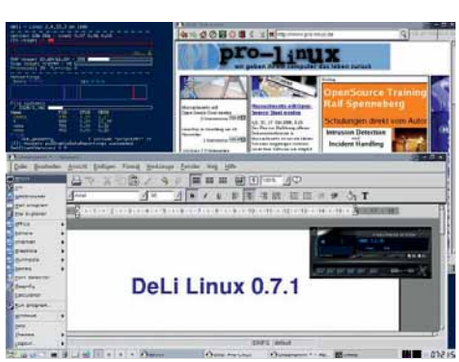
Мэри Перрин [Mary Perrin]

Похоже, инсталлятор не может опознать CD-привод: это бывает с нестандартными устройствами ATAPI IDE. Раз Вам понадобилась загрузочная дискета, скорее всего, так и есть. Невозможность набрать **C:** почти наверняка вызвана неверной раскладкой клавиатуры, из-за нее вводятся не те символы. Символ \ там есть, а чтобы его найти, просто переберите несколько клавиш, до получения нужного результата. Похожие проблемы были у меня с поиском символов # и @ при загрузке с Live CD, применявшего US-раскладку к моей UK-клавиатуре. Охоты за клавишами можно избежать, указав корректную клавиатуру на этапе инсталляции, иначе во время работы на DeLi Linux будут те же проблемы.

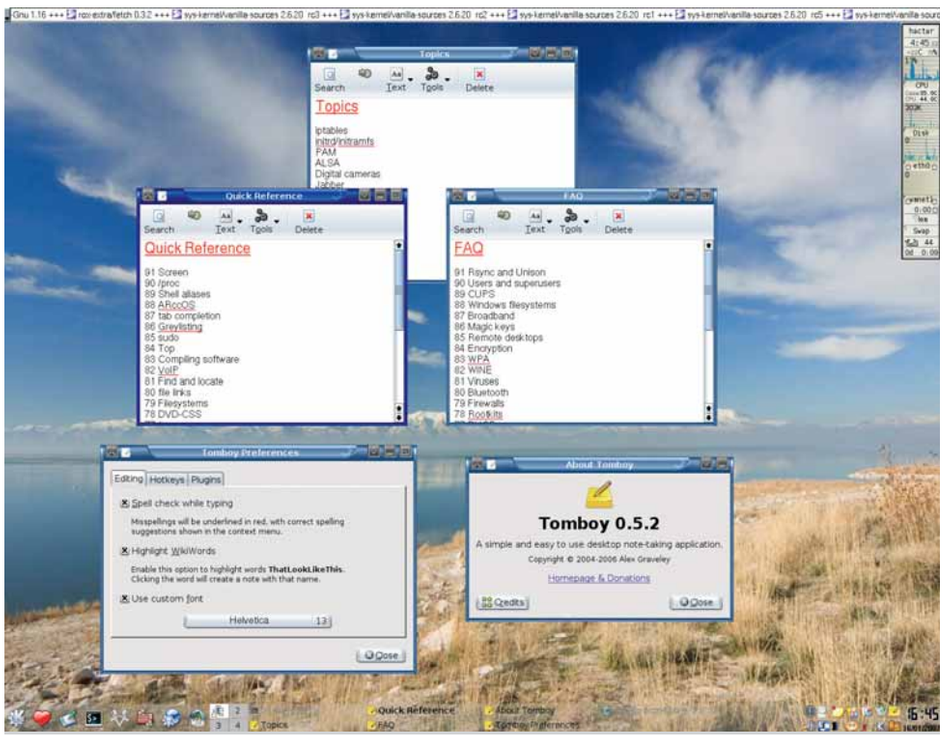
Удачи Вам с DeLi Linux: заставить хоть что-нибудь работать на 486-м далеко не просто. Могу посоветовать Damn Small Linux – еще один малый дистрибутив, его можно найти на диске LXF89. **МС**

10 Скан и обман

В SUSE 10.2 распознает мой «все-в-одном» принтер Photosmart и прекрасно печатает, а вот копировать не может. Единственный дистрибутив, который справляется с этим – Ubuntu, но я предпочитаю SUSE, он стабильнее на моей машине.



DeLi Linux любит старые машины, но не любит старые CD-приводы.



Tomboy неотъемлем от жизни, как тепло и кров – мы пользуемся им, готовя эти страницы.

Другой вопрос. *Tomboy* – отличная штука, но SUSE 10.2 с ним не справился из-за обычной ошибки Alt/F12. Я закрыл программу, но при попытке перезапуска мне было указано, что она уже работает. Удалил ее через *Yast*, а она никуда не делась. Загрузил новейшую версию с сайта *Tomboy* – она потребовала столько всего, что от нее пришлось отказаться. Жажду поработать с программой, но все мои попытки бесплодны.

joslam

За сканирование и копирование отвечает SANE. Один из способов заставить его работать на SUSE – это загрузиться в Ubuntu и узнать, какой драйвер там установлен:

```
scanimage --list-devices
```

Драйвер будет указан перед названием устройства. Например, мой сканер использует драйвер Genesys и показывает следующее:

'device `genesys:libusb:005:003' is a
Canon LiDE 60 flatbed scanner'.

Определить нужный драйвер – это пол-дела, а чтобы завершить дело, надо схитрить: скопировать файлы конфигурации (обычно они находятся в **/etc/sane.d**) из Ubuntu в SUSE.

Что Вы подразумеваете под «обычной» ошибкой Alt/F12, я не знаю. Я пользуюсь *Tomboy* уже давно и даже не слышал, чтобы сочетание Alt+F12 делало что-нибудь, кроме вызова меню, для которого оно и предназначено. Возможно, эту комбинацию заняло какое-нибудь другое приложение, но ведь горячие клавиши нетрудно изменить в диалоговом окне настройки параметров *Tomboy*.

Если программа не запускается или создает другие проблемы, первое, что нужно сделать – запустить ее не из меню, а в терминале. Проблему это вряд ли устранит, зато даст больше информации к размышлению. Похоже, что *Tomboy* остался работать в фоне. Это можно проверить, набрав

```
ps -ax | grep -i tomboy
```

в терминале. Будут показаны все процессы, в имени которых имеется слово «*Tomboy*». Крайний левый пункт в выводе будет идентификатором процесса (PID), который поможет убить процесс:

```
kill nnnn
```

где **nnnn** – PID процесса. Убедившись, что программа прекратила работу, Вы можете запустить ее снова в штатном режиме. **НБ**

11 Вон из моего порта!

В На моей материнской плате лишь один последовательный порт. А еще у меня есть Psion 5 и Garmin Geko 201, оба с последовательным подключением. Мне кажется, некоторые программы, связанные с Psion, работают в фоне, даже когда сам он отключен, и лишают Garmin (и программы, включая *GPSman*) доступа к последовательному порту.

Команда top в Linux (Kubuntu Edgy) не выявила известных мне приложений, связанных с Psion; не дала результатов и ps -aux. Не подскажете, какие процессы могут запирать порт при инсталлированном *KPsion*, и как их отстрелить, не удаляя пакетов *KPsion*?

Если ничего не поможет – думаю, придется просто купить PCI-карту с последовательными портами!

Мартин [Martin]

О А помните Unix-мантру «Все есть файл»? Она применима и здесь, так как последовательный порт можно расценивать как файл, а команда *lsof* показывает все открытые файлы. Без аргументов команда показывает все файлы, открытые системой, и Вы удивитесь, как много их на Linux (у меня сейчас открыто свыше 7 000). Круг поисков можно сузить, указав имя файла; в Вашем случае это будет, вероятно,

/dev/ttyS0, первый последовательный порт.

Взгляните на этот код:

```
sudo lsof /dev/ttyS0
COMMAND  PID USER   FD   TYPE DEVICE
SIZE NODE NAME
belkin 9758  nut    4u   CHR  4,64      5134 /dev/ ttyS0
```

Этот пример показывает, что мой порт занят монитором моего ИБП. В Вашем случае будет указана программа, запирающая порт и мешающая другим – вероятнее всего, она связана с *KPsion*. Возможно, она использует символьную ссылку на **/dev/ttyS0**, например, **/dev/modem** или **/dev/psion**. Увидеть их можно с помощью

```
ls -l /dev/ | grep ttyS0
```

Попробуйте пропустить через *lsof* и символические ссылки. Стоит взглянуть также на программы, запускаемые при загрузке. Ubuntu и другие производные от Debian по умолчанию используют runlevel 2, поэтому

```
ls -1 /etc/rc2.d
```

покажет все запущенные программы.

Если Вы хотите добавить порт, то дешевле (и проще) всего купить USB-адаптер. На eBay он стоит несколько фунтов и «просто работает», когда его подключают. Некоторые даже хвастают Linux-совместимостью – и на eBay тоже! **НБ**

12 Не задушишь, не убьешь

Пробую установить и настроить программу *DansGuardian*, описанную в журнале LXF81. Собрал версию 2.9.8.2 из исходных текстов. По умолчанию программа установилась в **/usr/local/**. Запускаю ее командой

```
# /usr/local/sbin/dansguardian start
```

после чего все прекрасно работает. Но вот в чем заковыка: больше ни на какие команды *DansGuardian* не откликается:

```
# /usr/local/sbin/dansguardian status
# /usr/local/sbin/dansguardian restart
# /usr/local/sbin/dansguardian stop
```

выдают одно и то же:

```
I seem to be running already!
```

Если «убить» программу командой ‘kill -9 PID’, она перестает запускаться даже после перезагрузки компьютера.

```
# /usr/local/sbin/dansguardian start
Error opening/creating log file. (check ownership and
access rights).
I am running as squid and I am trying to open /usr/local/
var/log/dansguardian/access.log
```

Мой дистрибутив – SUSE 10.2. Подскажите, в чем может быть дело? И еще – как настроить автоматический запуск DansGuardian при загрузке компьютера?

Игорь Редькин

Я бы порекомендовал Вам подойти к решению этой проблемы с другой стороны и установить готовый RPM-пакет *DansGuardian*. Его можно загрузить с домашней страницы проекта (http://dansguardian.org/downloads/2/Stable/SUSE/) или поискать в репозиториях Вашего дистрибутива. После того, как пакет будет установлен, вы сможете активировать автоматическую загрузку *DansGuardian* во время старта системы в разделе «Сервисы» *Yast*. **ВС**

13 Происки Microsoft?

Установил пакет **monodevelop-0.12-0.novell.noarch.rpm** с диска, прилагавшегося к январскому номеру **LXF**, в своей системе OpenSUSE 10.1. При попытке запустить *MonoDevelop*, получаю следующую ошибку:

```
System.Runtime.Remoting.RemotingException: Unix
transport error.

 Server stack trace:
 in <0x0008f> Mono.Remoting.Channels.Unix.
UnixMessageIO:ReceiveMessageStatus (System.IO.
Stream networkStream, System.Byte[] buffer)
...
```

Пожалуйста, помогите решить проблему, а то учебник Mono мне приходится изучать чисто теоретически!

temich_nd

Вам «повезло» столкнуться с достаточно известной проблемой взаимодействия между Mono 1.1.13 и Boo, Python-подобным языком программирования для среды .NET (см. LXF77). Ее можно решить тремя способами: обновить Boo до версии 0.7.5, Mono – до 1.1.14 и выше или просто удалить Boo-дополнение из MonoDevelop, воспользовавшись командой **mdtool gsetup**. Учитывая, что Boo вряд ли понадобится Вам в процессе чтения учебников по Mono (они ориентированы на C#), я бы рекомендовал третий путь – тем более, что он не требует лишних скачиваний. **ВС** LXF

Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.

Лучшие новинки открытого ПО на планете

LXF HotPicks

В ЭТОТ РАЗ ТОЛЬКО ДЛЯ ВАС: Qemulator » Klavaro » Bugle » Audacity » JSHoldem » Dagii » Breve » ExactImage » KX Generator » SystemRescueCD

Ричард Драммонд
Ричард – свободный разработчик, писатель и отец двух детей. Он живет в Индиане, США, где отчаянно скучает по британскому ТВ, подогретому пиву и сосискам.

Программа для запуска Qemu

Qemulator

Версия 0.3.2 **Сайт** http://qemulator.createweb.de

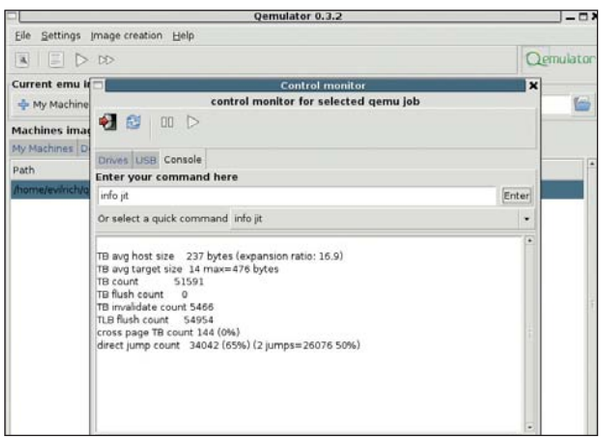
Qemu – солидный набор инструментов для эмуляции, способный виртуализовать не только главный процессор системы для эмуляции внешнего процессора, но даже интерфейс системных вызовов Linux. Все это здорово, но пользоваться им не так-то просто.

Поможет *Qemulator*, графический инструмент для настройки, запуска и управления сессиями *Qemu*: он предоставляет всю мощь *Qemu* в доступной для среднего пользователя форме. Главное окно отображает список «закладок» – именованных ссылок на сохраненные конфигурации *Qemu*. Вы просто выбираете одну из них и нажимаете кнопку Play для запуска *Qemu*-сессии с заданными настройками. В дополнение к этому, *Qemulator* поддерживает управление заданиями: запустите сессию (специальная кнопка, похожая на клавишу быстрой перемотки, находится справа от кнопки Play), и вы сможете приостанавливать ее, несильно менять настройки (например, перечень смонтированных или подключенных по USB устройств) и передавать *Qemu* команды «на лету».

Но главное преимущество использования *Qemulator* по сравнению с чистым *Qemu* – это простота создания конфигураций. Например, предоставляется инструмент для генерации образов дисков *Qemu* во всех поддерживаемых в настоящее время форматах. Образ используется для создания именованной конфигурации, в которой можно отрегулировать любые параметры: использование памяти, настройки сети или разрешение на доступ к USB-устройствам.

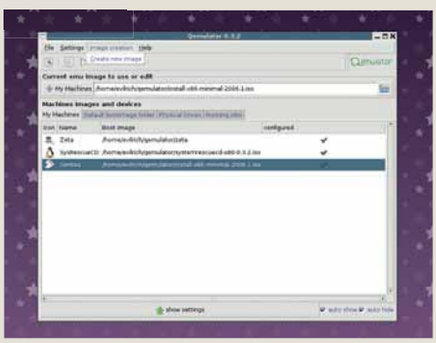
Qemulator написан на Python, так что его легко установить – он требует только Python 2.4 и зависит от *GTK* и *libglade*. Просто распакуйте пакет *Qemulator* и выполните **./setup.py** для запуска собственного графического установщика. Споткнуться он может разве что в системе без команды *sudo*: здесь потребуется запустить программу от имени суперпользователя.

Хотя комбинация *Qemulator* плюс *Qemu* уступает в лоске и простоте использования проприетарным решениям типа *VMware*, это ценное средство, если необходима виртуализация при скудном бюджете. Он несомненно будет улучшаться и, быть может, однажды превзойдет своих коммерческих собратьев.



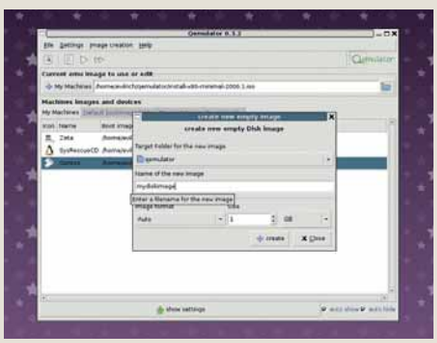
› Управление заданиями от *Qemulator* дает вам полный контроль над виртуальной машиной.

Шаг за шагом: Создание образа диска



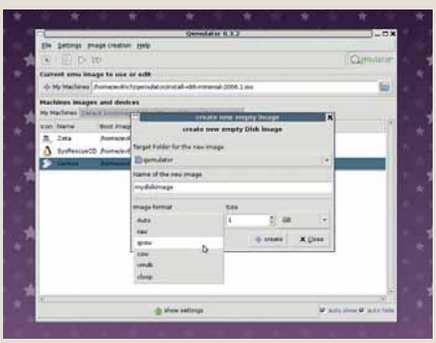
» Открытие диалога

Для создания образа жесткого диска сперва запустите *Qemulator*, затем выберите пункт Create New Image [Создать новый образ] меню Image Creation [Создание образов].



» Задание имени образа

В появившемся диалоге выберите каталог, в котором создастся образ (проще всего оставить все по умолчанию) и введите имя файла.



» Задание формата и размера

Теперь выберите формат файла – вероятно, лучшим выбором для файловых систем Linux будет RAW – а затем размер. Щелкните на Create [Создать] и наслаждайтесь!

Тренажер клавиатуры

Klavaro

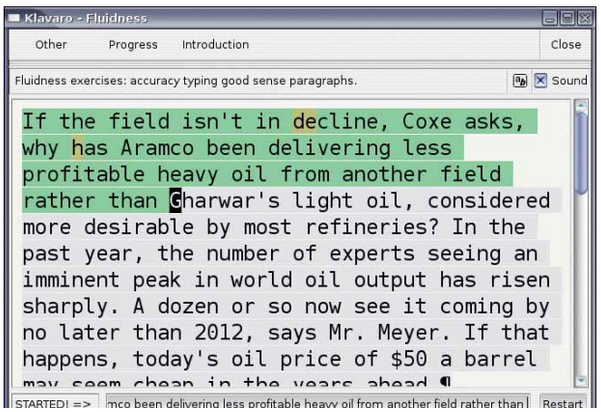
Версия 1.0.1 **Сайт** http://klavaro.sourceforge.net

Если у вас в кладовке не припрятан HAL 9000 на замену вашему персональному компьютеру, то лучший способ ввести текстовые данные в настольный ПК – это воспользоваться старой доброй клавиатурой; конечно, если вы умеете ею пользоваться. К сожалению, набор текстов не есть умение, достижимое за одну ночь. Чтобы перейти от двухпальцевого, извините за выражение, «метода» к набору 100-слов-в-минуту-вслепую, нужна практика. Много практики.

Klavaro – тренажер клавиатуры; он пытается научить вас основам слепой печати и предоставляет комфортное окружение для оттачивания навыков. Здесь четыре уровня упражнений, с нарастающей сложностью. Первый уровень знакомит с расположением клавиш. Вы вводите группы случайных символов, начиная с символов, расположенных на «опорных» клавишах (типа ФЫВА и ОЛДЖ в русской раскладке).

По мере вашего прогресса, *Klavaro* добавляет все больше и больше прочих клавиш. Упражнения второго уровня проверяют приспособляемость. Как и на первом уровне, надо вводить группы случайных символов, но используется уже вся клавиатура, включая цифры, знаки пунктуации и заглавные буквы. На третьем уровне вводятся произвольные слова и проверяется ваша скорость, а на четвертом (последнем) уровне – целые абзацы текста, а также проверяется скорость и плавность набора.

После каждого упражнения *Klavaro* выводит оценку точности набора, а на более высоких уровнях – скорости и плавности. Если вы не удовлетворите требованиям к точности и скорости данного уровня, *Klavaro* посоветует его повторить. Прилагаются также графики для визуализации вашего прогресса.



› ***Klavaro* содержит абзацы полноценного текста для тренировки, но тематика их часто скучна.** [На рисунке приведено обсуждение нефтедобычи с аналитиками, – прим. пер.]

«Klavaro выделяется среди прочих своим интернационализмом.»

Среди аналогичных программ *Klavaro* выделяется интернационализмом. Он предусматривает настроенные стандартные US, British, French, German и многие другие раскладки клавиатуры, но можно определить и свою собственную; он содержит список слов и текстовых абзацев для поддерживаемых локалей, но опять же легко добавить свои собственные (это обычные текстовые файлы).

Авторы создали RPM- и Debian-пакеты, но *Klavaro* проще простого собрать самому (он требует только *GTK*). Кому не до машинописных курсов, тем *Klavaro* очень пригодится – мы используем его ежедневно, и аш набор безупречен!

Отладчик OpenGL

Bugle

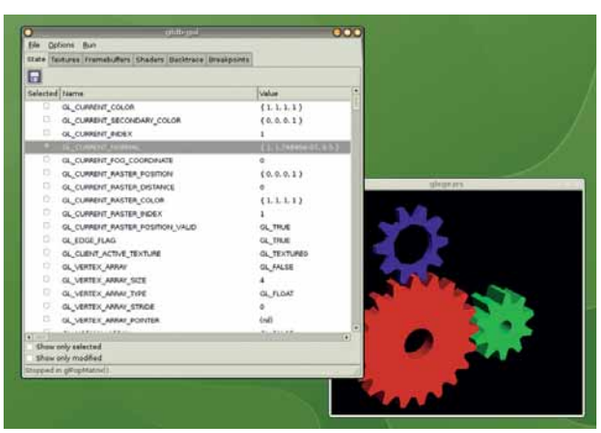
Версия 20070107 **Сайт** http://bugle.sourceforge.net

Любой работяга скажет вам, что главное – выбрать правильный инструмент. Например, попытка забить гвоздь отверткой отнимет много времени, а может отнять и пальцы. То же самое – при отладке приложений: использование подходящего инструмента делает работу легче и быстрее. Если вы анализируете приложение OpenGL, *Bugle* – набор инструментов для отладки OpenGL – сэкономит вам кучу времени и избавит от головной боли.

Ядро *Bugle* – разделяемая библиотека, которую ваша программа может предварительно загружать для перехвата вызовов OpenGL. По умолчанию эта библиотека просто пропускает их, но можно настроить *Bugle* на использование набора элегантных инструментов отладки, от простого журналирования всех вызовов OpenGL до захвата видео с GL-выхода. Все эти функции реализованы в виде фильтров и настраиваются в простом текстовом файле. Пример файла конфигурации поставляется с дистрибутивом *Bugle*, но если вы пожелаете создать свой собственный, то документация превосходная.

Кроме библиотеки, *Bugle* включает графическую и командную оболочки, значительно ускоряющие и упрощающие отладку. Например, чисто текстовая *gldb* может автоматически загружать библиотеку *Bugle* и запускать вашу программу. Для остановки и возобновления выполнения, задания точек останова в OpenGL-функциях, запроса состояния OpenGL, создания снимков экрана, разрешения или запрещения любых, определенных *Bugle*, фильтров и т.д. предоставляются простые текстовые команды в духе GDB (отладчика GNU) Можно даже следить за стеком или запускать сам GDB.

Оболочка *Gldb-gui* делает все это и многое другое в приятном графическом окружении, построенном на *GTK*. С *Gldb-gui* легко можно прерывать выполнение и визуализовать



› ***Bugle* предоставляет несколько способов опроса состояния *OpenGL*.**

«Можно настроить Bugle на использование набора инструментов отладки.»

содержимое буферов переднего и заднего плана и глубины, выводить список переменных состояния OpenGL, проверять текстуры, просматривать список шейдеров и задавать точки останова.

На данный момент нет двоичной версии *Bugle*, доступной для загрузки, так что вам придется собрать его самостоятельно из исходных текстов. Сделать это не сложно: только вызвать **configure** и **make**; но вам потребуются *GTK* и *GTKGLExt* для сборки графического интерфейса и *FFmpeg* для записи видео. *Bugle* дает возможность запросто представить, что ваша программа делает в каждый момент.

Звуковой редактор

Audacity

Версия 1.3.2 beta **Сайт** http://audacity.sourceforge.net

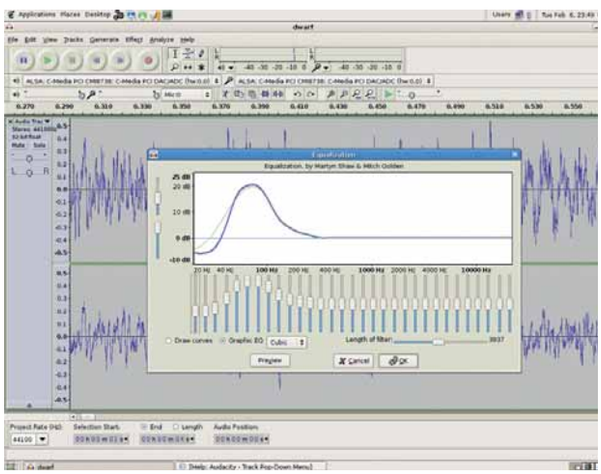
Мы рассматривали *Audacity* ранее в разделе HotPicks LXF54, учебниках по работе со звуком и нескольких обзорах. Не удивляйтесь, что мы обращаемся к нему вновь: он до того хорош, что устоять невозможно.

Audacity – прекрасный, хорошо оснащенный звуковой редактор. Его графический интерфейс проворен и легок в использовании, и может с равным успехом применяться и для простейших задач, и в профессиональной сфере. Имеется две версии *Audacity*: стабильная и менее зрелая бета-версия (мы рассмотрим последнюю). Обе версии в основном используют одинаковые формы и функции, но бета-версия построена на основе новейшего графического инструментария *WXGTK* и, по сравнению со старой, имеет более ухоженный вид (например, сглаживает шрифты), да и вообще предоставляет более приятный и опрятный интерфейс пользователя. Отрицательный момент, кроме нестабильности, в том, что набор двоичных пакетов бета-версии весьма скуден, и если вы захотите попробовать приложение, придется собрать его из исходного текста самостоятельно.

Полная функциональность *Audacity* собирается по различным библиотекам от третьих лиц, так что самостоятельная компиляция непроста. Зависимости в какой-то мере разрешены путем включения некоторых библиотек в архив *Audacity*, например, *PortAudio* для вывода звука, *libsndfile* для импорта и экспорта, и так далее – но вам точно потребуется последняя сборка *WXGTK* и поддержка желаемых аудио-форматов (таких, как OGG), и *libmad* для MP3. Выполните **./configure**, затем прочитайте вывод и проверьте, что все зависимости разрешены. Но как только версия 1.3 *Audacity* станет стабильной, и будет доступен двоичный пакет, большинство пользователей сможет избавиться от этих мук.

Звуковые байты

Audacity – многодорожечный звуковой редактор. Вы можете импортировать аудио-данные из любых стандартных форматов или записывать их напрямую с любого источника звука, поддерживаемого вашей звуковой картой; *Audacity* может даже синтезировать простые звуковые сигналы самостоятельно. Он управляет моно- или стерео-сигналами, и все загруженные в данный момент сигналы отображаются на экране. По умолчанию, звуковые отсчеты имеют внутреннее представление в виде 32-битного числа с плавающей точкой, для точности, но это, а также различные параметры экспорта, можно изменить в настройках. Можно также выбрать частоту дискретизации для всего проекта.



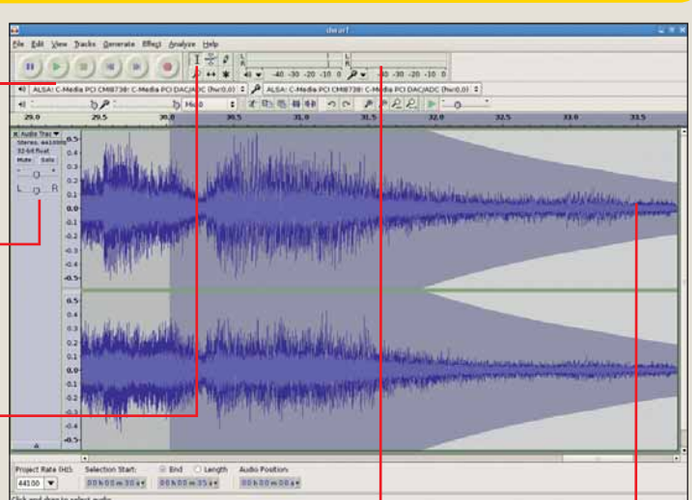
› ***Audacity*** **набит расширениями для любых преобразований, вроде этого эквалайзера.**

Главное назначение *Audacity* – редактирование, а не приятный вид, и для этой цели предусмотрен вагон инструментов. Вы можете объединять и разбивать дорожки, вырезать, копировать и вставлять части, микшировать дорожки и многое другое. Панель инструментов позволяет сдвигать дорожки по времени, увеличивать и уменьшать громкость во времени и даже редактировать сэмплы вручную. Недавние изменения могут отображаться в диалоге, позволяющем отменить ряд манипуляций, сделанных за последнее время.

Наиболее мощные инструменты, предлагаемые *Audacity*, подключаются в виде расширений. Поддерживается два типа расширений: расширения LADSPA (*the Linux Audio Developer's Simple Plugin API – см.* www.ladspa.org) и расширения, написанные на языке программирования Nyquist, диалекте Lisp (*см.* www.cs.cmu.edu/~music/music.software.html). Хотите – используйте Nyquist для создания собственных расширений, но стандартный набор, включенный в *Audacity*, предоставляет столько эффектов, что большинству пользователей их хватит надолго. Примеры включают настраиваемый БПФ-фильтр, фазовращатель, программу сжатия и эффект «квакушки». Более практичные расширения выполняют такие задачи, как удаление шума щелчков и нормализация.

В целом, *Audacity* – впечатляющий пакет. С его продвинутыми инструментами и расширяемостью он обладает колоссальной мощью, оставаясь доступным обычному пользователю. Более того, прилагаемая документация и online-учебники помогают новичкам моментально освоиться с приложением. Ждем выхода стабильной версии 1.3!

Изучаем интерфейс Audacity



Панель управления
Знакомые кнопки, и делают что ожидается: воспроизведение, паузу, остановку и запись звука.

Инструменты дорожек
Выпадающее меню содержит инструменты для изменения способа отображения звука, а также конвертирования формата и скорости сэмпла.

Инструменты редактирования
Здесь расположены основные инструменты редактирования *Audacity*, включая масштабирование, сдвиг по времени и инструмент огибающей (позволяющий редактировать громкость).

Индикаторы
Индикаторы входного и выходного уровня позволяют легко обнаружить отсечку звуковых данных.

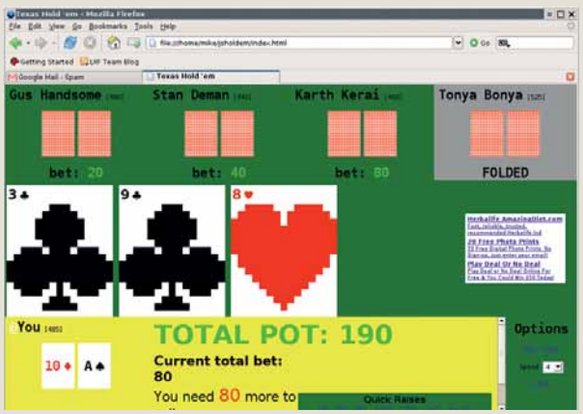
Визуализация дорожек
Звуковые волны можно отображать в виде зависимости амплитуды от времени с настраиваемым пользователем увеличением.

HotGames Развлекательные приложения

Карточная игра

JSHoldem

Версия 0.94 beta **Сайт** http://jsholdem.sourceforge.net



› По крайней мере, здесь вы не сможете в отчаянии поставить на кон вашу машину/дом/жену/любимую приставку SNES...

Перед вторым кругом у вас на руках десятка и валет одной масти, поэтому вы удваиваете втемную и прикупаете. Приходят валет, король и девятка, вызывая бешеный вихрь мыслей. Что делать? Налицо стрит, и имеется шанс на флеш – рискнуть или просто поставить на пару и надеяться, что ваша сдача достаточно хороша, чтобы кто-то рискнул перебить? Возможно, даже придет трио (prile – прайл)...

Если для вас это звучит полной белибердой, но вы заинтригованы техасским покером «холдем» (в конце концов, это хороший баланс удачи и умения, а играть ужасно интересно), поможет *JSHoldem*. Его даже не нужно устанавливать; *JSHoldem* написан на HTML и JavaScript, так что просто распакуйте **jsholdem-0.94-beta.zip** и откройте в вашем браузере **index.html** из получившегося каталога.

Хотя графика его напоминает детские рисунки мелом, *JSHoldem* имеет солидный AI для четырех компьютерных игроков – они не просто играют в карты, но также блефуют и пытаются заставить вас спасовать. Вы начинаете с 500 единиц неизвестной валюты, и можете делать ставки от 20 до 120 единиц, или, если чувствуете полную уверенность в удаче, можете идти ва-банк.

JSHoldem норовит взвинтить напряжение, используя паузы (после раздачи карт или хода соперника), но если они вас раздражают, можете увеличить скорость игры. Однако игра не совершенна: когда компьютерные игроки меняют тактику, их очень легко подбить на большие ставки, что позволяет выигрывать много разом, без нудного накопления.

> «Четыре компьютерных игрока блефуют, пытаясь заставить вас спасовать.»

Компьютерный покер никогда не станет столь же затягивающим, как реальная игра – ведь не видно выражений лица соперников и не слышно их шуточек, но *JSHoldem* отлично работает, если учесть, что это даже не отдельная программа. Акулам блефа он может показаться слишком легким – но если вы хотите научиться играть, то найдете его весьма полезным.

Движок-интерпретатор сюжетных игр

Dagii

Версия 0.3 **Сайт** www.dagii.org

› Что может быть лучше гуляния с вашей свинкой в лесу, полном драконов? *Black Cauldron в Dagii*.

Давным-давно, когда ПК были юными, Sierra Online произвела революцию в компьютерных играх, создав графическую сюжетную игру (жанр «Adventure»). По сегодняшним меркам, ранние игры Sierra выглядели просто чудовищно, но в них все еще интересно играть. Желающим окунуться в историю игр рекомендуем *Dagii* – интерпретатор, позволяющий играть в классику, включая *King's Quest* и *Space Quest* под Linux.

Если это ваша страсть, то вам знакомы и другие AGI-интерпретаторы для Linux, например, *Nagi* и *Sarien*. Увы, работа над обоими в последние годы приостановилась. Движок Sarien ввели в популярный и чрезвычайно неудачно названный проект *ScummVM*, но релизы с поддержкой AGI еще не готовы. В данный момент *Daggi* завершен в меньшей степени, чем эти ранние движки, и обеспечивает худшую совместимость, зато по крайней мере активно развивается. Вдобавок он уже обзавелся собственными уникальными функциями, выделяющими его на фоне ранних предложений: скажем, включены опция создания снимков экрана и инструменты для тех, кто создает или отлаживает AGI-игры, например, встроенный просмотрщик спрайтов.

Разработчик *Dagii* Дэвид Саймондс [David Symonds] пока не создает двоичные пакеты для загрузки; придется вам закатать рукава и собрать его вручную. Требуются только SDL и *libpng* – они, вероятно, у вас уже имеются. Никаких необычных опций компиляции нет, так что выполните **./configure**, **make dep** и **make**. После запуска *Dagii* ищет файлы с игровыми данными в текущем каталоге и каталоге **/usr/share/games/sierra**. Переопределить этот путь можно, указав новый в командной строке или при помощи опции gameselecter.search_paths в файле конфигурации.

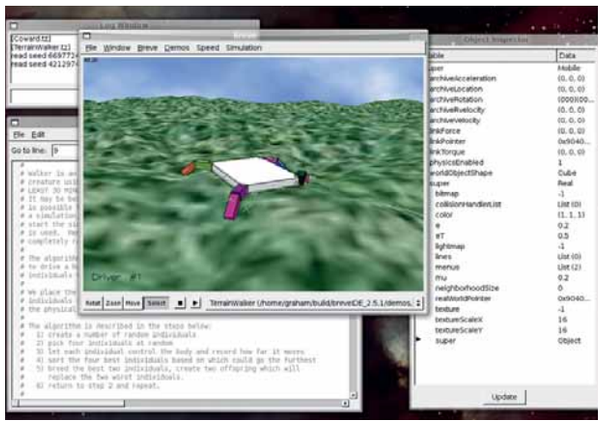
> «Играйте в классику, включая King's Quest и Space Quest под Linux.»

Dagii весьма изящно закрыл брешь. Ну да, звука в нем пока не предусмотрено, работает он только с версиями игр Sierra для PC и не поддерживает никаких самопальных AGI, но не так долго всего этого ждать. А теперь, если вы нас извините, мы прогуляем любимую свинку...

Симулятор искусственного интеллекта

Breve

Версия 2.5.1 **Сайт** www.spiderland.org



После ночи расчетов, эта имитация учит ходить кубоид с четырьмя конечностями.

Искусственный интеллект (ИИ) – обоюдоострый меч. Это волнующий и чрезвычайно сложный предмет, вотчина академиков и разработчиков игр. Продукт диссертации Йона Кляйна [Jon Klein], *Breve*, не исключение. Он делает изучение ИИ не легче, но забавнее.

Вам не нужно знать о брайтенберговской [V.Braitenberg] модели Роя, чтобы получить удовольствие от этого приложения. Чтобы увидеть ИИ в действии, просто запустите приложение и выберите меню Demo. Вы увидите 3D-имитацию выбранной модели, вместе с текстовым окном, отображающим ее исходный код. Для начала представления просто щелкните на кнопке Play.

Некоторые демонстрации удивительны. Полюбуйтесь, как кубоид с четырьмя конечностями пытается встать на ноги и пройтись по холмистой местности, или посмотрите знаменитую игру *Жизнь* в трехмерном варианте. Еще одна демонстрация направляет случайное скопление птиц (отображаемых на экране конусами) в естественный полет, наблюдаемый у стай скворцов. Минут через десять он становится пугающе натуральным. Просто глядя на экран, вы начинаете понимать, как каждая птица находит свое место в постоянно движущейся массе виртуальных пернатых. Без подобных инструментов, для получения того же уровня понимания мы были бы обречены на чтение *The Emergence of Collective Behaviour in Evolving Populations of Flying Agents* [Возникновение коллективного поведения в эволюционирующей массе летающих агентов].

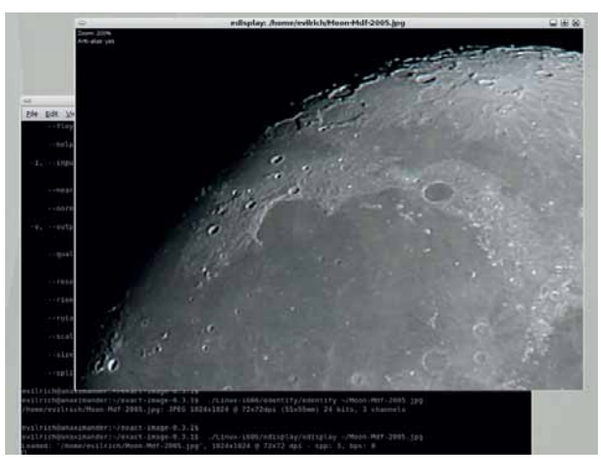
Если вы хотите с чем-то повозиться, то у вас есть только Steve, Стив – это имя, данное объектно-ориентированному языку программирования, использованному для создания симуляций *Breve*. Он поразительно прост для понимания: лучший способ начать – переделать один из множества демо-примеров. Но даже не тронув ни строчки, вы можете взаимодействовать с 3D-моделями, перемещая объекты и меняя различные параметры. К основному приложению добавляются расширения, пригодные в любой имитации: хороший пример – MIDI-плагин, связанный с имитацией Роя, он посылает ноты на синтезатор. Online-документация также превосходна, содержит различные учебники и даже курс для углубленно изучающих материал. *Breve* – луч света в темном царстве, без которого многие из нас так и сидели бы в потемках.

> «Breve не облегчает изучение ИИ, но делает его забавным.»

Пакет для обработки изображений

ExactImage

Версия 0.3.1 **Сайт** www.exactcode.de/oss/exact-image



Просмотрщик изображений *ExactImage*, *EDisplay*, использует ускорение OpenGL для молниеносного масштабирования и фильтрации изображений.

Немецкий магазин Linux-программ ExactCODE пытается внести популярный пакет обработки изображений *ImageMagick* в список ископаемых, заменив его более новым и более эффективным *ExactImage*. Работа над ним не завершена, и он не поддерживает такого количества форматов, как конкурент – *ImageMagick*, но скоро добавятся новые.

API *ExactImage* выполняет обычные функции обработки изображений – преобразование цвета, масштабирование, вращение – и функцию поинтереснее: ускорение для кодеков. Например, JPEG-кодек может выполнять вращение JPEG-изображений без потери качества и быстро уменьшать размеры картинок.

Кроме API, *ExactImage* содержит набор инструментов, похожих на имеющиеся в *ImageMagick*. Инструмент *Edentify* копирует *ImageMagick*'овский *identify*, показывая информацию об изображении: формат файла, размеры и детали цветового пространства. Более подробный вывод пока не поддерживается. Затем идет *Econvert*, выполняющий работу *convert* и способный изменять формат файла или глубину цвета, вращать, переворачивать или масштабировать изображение.

Компонент просмотра изображений, *Edisplay*, бросится в глаза пользователям как главным козырь по сравнению с *ImageMagick*. Хотя на данный момент это просто просмотрщик – с ним нельзя трансформировать изображения и сохранять результаты, как с *display* – он намного, намного быстрее. Изображения также можно быстро масштабировать (со сглаживанием или без) и панорамировать при помощи мыши.

Двоичные пакеты пока не доступны для загрузки: вам придется самостоятельно собрать его из исходных текстов. Зайдите на http://enlightenment.freedesktop.org, возьмите последние архивы *libeet* и *libevas*, загрузите их, скомпилируйте и установите. Затем можно компилировать *ExactImage* – простыми **./configure** и **make**.

Хотя *ExactImage* устанавливается долгонько, мы думаем, что он того стоит – даже на столь ранней стадии разработки. Разработчикам и пользователям (особенно тем, кому нужно манипулировать множеством изображений) пригодятся его скоростные функции обработки изображений, да и функции вроде вращения JPEG-снимков без дополнительных искажений тоже очень украсят их жизнь.

> «Есть обычные функции обработки изображений и ускорение для кодеков.»

Спасательный диск

SystemRescueCD

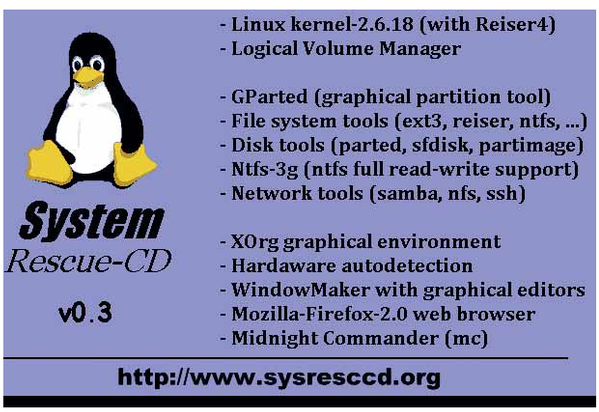
Версия 0.3.2 **Сайт** www.sysresccd.org

Хотя вашего внимания пытается добиться масса спасательных дисков, одним из наиболее продвинутых является *SystemRescueCD*. Его размер близок к 130 МБ, зато вы получаете массу удовольствия от этих многочисленных байтов. Можете выбрать традиционный способ: просто прожечь образ на диск и использовать его; или создать свой собственный образ; или даже установить и использовать его на USB-брелке, если пожелаете. Online-документация и сильное сообщество помогут, если вам потребуется нечто более экзотическое.

Хотя это всего лишь минималистичный LiveCD-дистрибутив, *SystemRescueCD* предлагает в приглашении загрузки кучу других возможностей: невероятно полезный *memtester*, жизненно важный при определении поврежденной памяти; Live-дистрибутив FreeDOS, если вам нужна совместимость с DOS; и *Aida* – инструмент диагностики оборудования. Естественно, имеется и Linux, на базе Gentoo.

SystemRescueCD поддерживает все стандартные файловые системы Linux, а основанные на Fuse инструменты NTFS-3G предоставят полный (чтение/запись) доступ к файловым системам NTFS. Для управления разделами имеется *Parted*; для резервирования – *Partimage*. Вы также найдете полный набор сетевых инструментов, включая поддержку Samba и ресурсов NFS. Далее, имеется полный дистрибутив *X.org* вместе с *WindowMaker* плюс *Firefox 2.0*, *GParted* и текстовый редактор *GVim*. Прямо-таки швейцарский перочинный нож среди спасательных дисков!

› Не позволяйте очаровашке-Туксу обдурить вас. SystemRescueCD намного мощнее вашего среднего Live-дистрибутива.



Инструмент настройки X

KX Generator

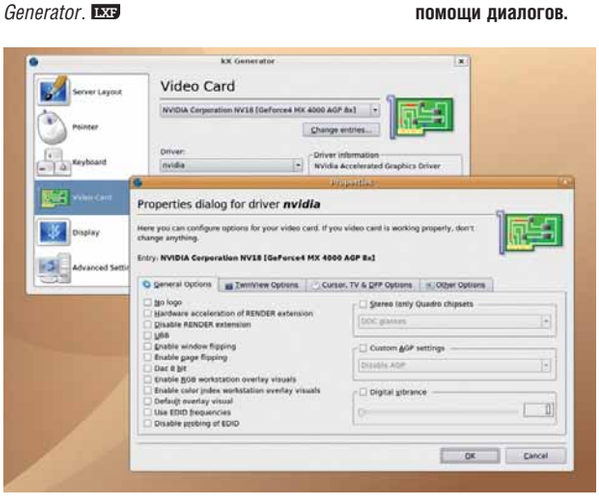
Версия 0.3.7 **Сайт** http://kxgenerator.xt.pl

Принимая настройку *X* по умолчанию в вашем дистрибутиве, вы вряд ли реализуете все возможности оборудования, особенно если хотите работать с несколькими мониторами. В этом случае можно или пойти тернистым путем самостоятельной настройки *X*, или обратиться к *KX Generator*.

KX Generator создан для KDE, но требует только базовые библиотеки, так что не пугайтесь, если вы KDE-фоб. Чтобы полностью задействовать инструменты графических режимов *KX Generator*, потребуется также утилита *GTF*, используемая для расчета режимов X.

Через серию диалогов можно настроить любой аспект вашей *X*-конфигурации, включая основные пункты: выбор драйвера для видеокарты, мыши и клавиатуры, редактирование атрибутов экрана и создание серверного окружения. Мы сочли наиболее полезной функцией *KX Generator* поддержку более продвинутых опций настройки, на которые способны *X*-драйвера и которые трудно припомнить, не закапываясь в man-страницы. Пока что охвачены далеко не все драйвера: поддерживаются несколько видеодрайверов и унаследованные драйвера клавиатуры и мыши. Когда их поддержка обрастет мясом, любители насладиться подстройкой своих систем вряд ли смогут обойтись без *KX Generator*. LXF

› Подстройте специальные опции для некоторых *X*-драйверов, среди которых проприетарный драйвер Nvidia, при помощи диалогов.



Также выпущены

Новые и обновленные программы, заслуживающие вашего внимания...

AnjutaIDE 2.1.0 Интегрированная среда разработки для GTK/Gnome www.anjuta.org

Bacula 2.0.1 Сетевая система резервирования www.bacula.org

DejaVu fonts 2.14 Популярные шрифты TrueType http://dejavu.sourceforge.net

DrawPile 0.3.0 Программа коллективного рисования http://drawpile.sourceforge.net

Fiwix 0.3.0 Свободное, совместимое с Linux ядро для целей обучения www.fiwix.org

Grsync 0.5.2 Графический интерфейс для инструмента синхронизации каталогов rsync www.opbyte.it/grsync

H2 1.0/2007-01-17 Быстрый встраиваемый движок базы данных на чистой Java www.h2database.com

HDparm 6.9 Инструментарий для работы с параметрами жестких дисков ATA http://sourceforge.net/projects/hdparm

KPlayer 0.6 Медиа-плейер для KDE на базе MPlayer http://kplayer.sourceforge.net

LinuxBIOS 2-2536 Замените вашу BIOS открытой, основанной на Linux альтернативой www.linuxbios.org

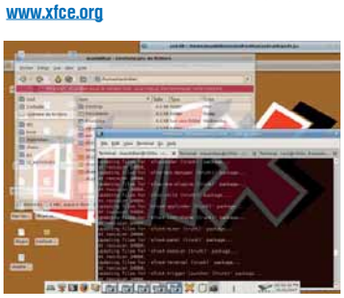
Mol 0.9.72_pre2 Новая жизнь клона виртуализации PPC http://mac-on-linux.sourceforge.net

Photo Organizer 2.33b Менеджер фотографий http://po.shaftnet.org

Stella 2.3.5 Соблазнительный эмулятор Atari 2600 VCS http://stella.sourceforge.net

› *Stella*: возврат к видеоиграм 70-х.

Xfce 4.4.0 Легковесный рабочий стол, теперь с новым файловым менеджером www.xfce.org



› Крутые эффекты с *Xfce*.

LXF DVD84

Думали, лучше Xgl ничего нет? Поглазейте-ка на это…

Майк Сондерс любовно подбирает содержимое диска Linux Format, а также поддерживает сайт www.linuxformat.co.uk.

И мило, и практично

Что важнее: внешний глянец или функциональность? Когда в 2001 году появилась Mac OS X, многие злопыхатели глумились над ее блестящим графическим интерфейсом, отметая виджеты и плавные тени как архитектурные излишества. И наоборот, сторонники Mac заявляли, что привлекательный интерфейс улучшает рабочую среду, потому что лучше воспринимается глазом, чем серые рабочие столы прошлого.

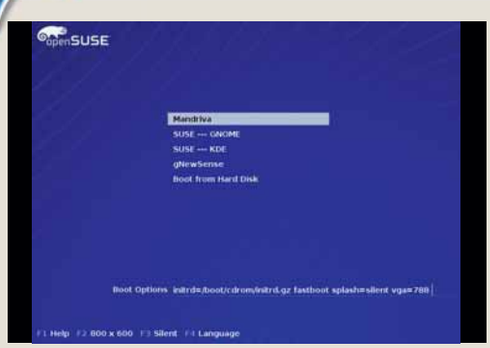
Похожие дебаты развернулись и в сообществе Linux после выхода *Xgl* и *Compiz*. Вращающийся 3D-куб, может, и красив, но способствует ли он хорошей работе? Лучше ли миниатюры окон, чем панели задач? Вправду ли отбрасываемые тени выделяют окна, или только создают корявый полу-трехмерный вид?

Пока все прочие затопляли Usenet этими разборками, разработчики трехмерного рабочего стола *Metisse* пошли другим путем: они поразмыслили о том, что люди делают на своих компьютерах, и создали эффекты рабочего стола в соответствии с этим. Вместо подхода, практикуемого *Xgl*, по типу: «Во какой эффект ряби на воде! Давайте его добавим!», команда *Metisse* рассуждает: «Как бы пользователям запускать и просматривать больше программ одновременно?»

Metisse вносит и свою лепту бессмысленных завитушек, но его технология и то, как он раскручивает рабочие столы Linux, произвели на нас сильное впечатление. А всего лучше, что вы можете сами опробовать его прямо с DVD этого месяца, благодаря новой Live-версии Mandriva, в которую включен *Metisse* – просто вставьте диск в привод, перезагрузитесь, и готово! На DVD этого месяца много чего интересного и достойного ознакомления, включая OpenSUSE 10.2 Live и уйму полезной документации.

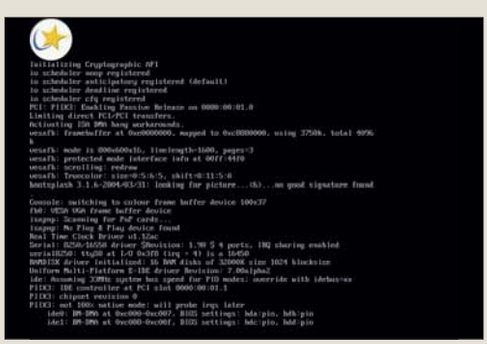
mike.saunders@futurenet.co.uk

Шаг за шагом: Изучаем функции Metisse



1 Загрузим

Загрузите ваш ПК с DVD, и нажмите Enter в меню загрузки (если у вас появится только текстовая строка, наберите Mandriva и нажмите Enter). Начнется загрузка Mandriva One 2007 Metisse.

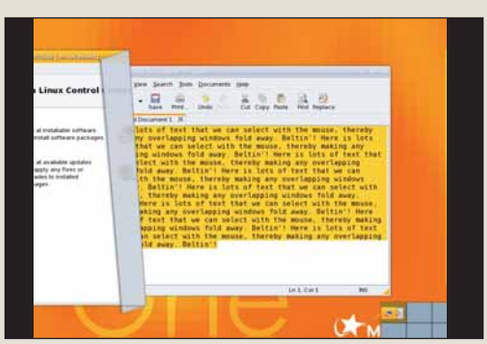


2 Проверим

Mandriva должна загрузиться без всяких проблем. Это может занять пару минут. Можете игнорировать бегущие сообщения, но если процесс остановится, внимательно прочтите информацию о произошедшей ошибке.

5 Облупим

Дважды щелкните и зажмите левую кнопку мыши на границе окна, чтобы увидеть кнопки «сворачивания»; двигайте мышью, чтобы «отогнуть» край окна.



6 Откроем

Откройте текстовый редактор, напечатайте несколько слов, затем перекройте окно редактора другим окном. Выделите мышью текст в окне редактора, и перекрывающее окно «отогнется»!

Дистрибутив Linux

Mandriva One Metisse

Возьмите Mandriva, один из дружелюбнейших дистрибутивов Linux в мире, переведите его в Live-формат, добавьте великолепия графического интерфейса пользователя *Metisse* – и у вас получится отличный дистрибутив. Мы включали полную версию Mandriva Free 2007 на DVD LXF87/88; в этом месяце у нас – Mandriva One, она работает прямо с DVD, не требуя установки, и вдобавок у нее в рукаве убойный козырной туз – новый графический пользовательский интерфейс *Metisse*. Подобно *Xgl* и *AIGLX*, *Metisse* добавляет на рабочий стол Linux 3D-рендеринг, но здесь это не просто красивость – эффекты максимизируют пространство экрана и способствуют производительности. См. наше пошаговое руководство ниже.

› Окно вверх ногами, вращается и сворачивается – пользы никакой, зато круче крутого!

Mandriva One дает также превосходный способ опробовать Linux, не устанавливая его на жесткий диск. Если Linux еще не установлен у вас на компьютере, вы можете загрузить его и увидеть, хорошо ли он работает (а заодно проверить, поддерживается ли ваше оборудование). Можете также дать его потенциальным пользователям Linux, чтобы они могли узнать из первых рук, почему Linux является превосходной альтернативой Windows (команда Mandriva включила в этот дистрибутив *OpenOffice.org 2.0.4* и *Firefox 1.5*). Не забудьте, что, поскольку он работает с DVD, вам надо сохранять все создаваемые файлы на внешнем устройстве, например, на USB-брелке.

Для запуска Mandriva One, загрузите ваш компьютер с DVD и нажмите Enter в меню загрузки. Минимальные требования – 256 МБ ОЗУ и процессор 1 ГГц, но для гладкой работы лучше иметь 512 МБ и 2 ГГц. *Metisse* – открытое ПО, основанное на почтенном оконном менеджере *FVWM* – для более подробной информации см. http://insitu.lri.fr/Metisse и www.mandriva.com/en/projects/metisse. А если на вашем компьютере есть только CD-ROM, можете создать CD версию Mandriva One с нашего DVD: подробности – в **index.html** на диске.

› Пусть супер-пупер-эффекты – это не ваше, все равно Mandriva очень полезна как офисная система.

3 Поедем

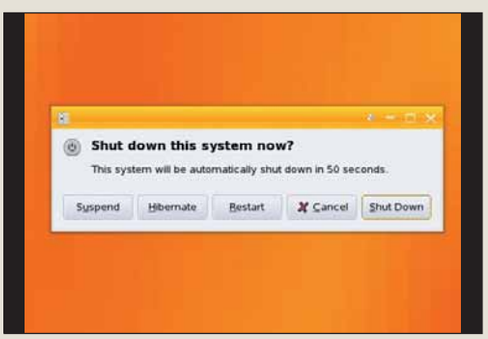
Перед показом рабочего стола Gnome Mandriva задаст вам несколько вопросов о конфигурации. Чтобы начать работу, откройте меню Applications, расположенное вверху слева.

4 Вертим

Нажмите и придержите левую кнопку мыши на границе окна – появится меню манипуляций. «Протащите» мышью до кнопки Y Rotate, чтобы прокрутить окно вокруг своей оси.

7 Настроим

Для изменения настроек Mandriva зайдите в Applications> System >Configure Your Computer (это Live-дистрибутив, так что они не сохранятся после перезагрузки).



8 Завершаем

Поработав в Mandriva, нажмите меню System, а затем – Shut Down, чтобы выключить или перезагрузить компьютер.

Дистрибутив Linux

OpenSUSE 10.2 Live

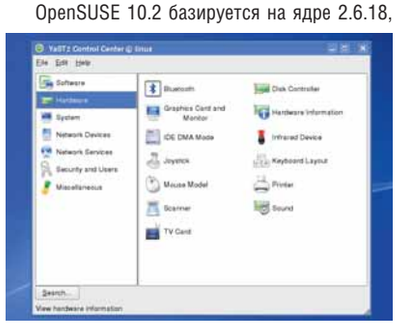
Хотя благодаря рабочему столу *Metisse*, опробовать Mandriva One весьма любопытно, тем, кто ищет в Live-дистрибутивах нечто посущественнее, нужен OpenSUSE 10.2. SUSE Linux остается одним из наиболее отлаженных и полнофункциональных дистрибутивов Linux, выпущенных за последнее десятилетие, и после перехода во владение Novell он начал широко использоваться на предприятиях; а проект OpenSUSE вовлекает в процесс разработки дистрибутива все более активное сообщество.

Поэтому мы рады предложить вам в этом месяце Live-версию OpenSUSE 10.2. Мы включали устанавливаемую версию дистрибутива два выпуска назад – в LXF89 – и если вы тогда его не опробовали, вот вам отличная возможность ознакомиться с дистрибутивом без его установки на жесткий диск. Даже если вы вполне довольны своим дистрибутивом, почему бы не иметь такой резерв на случай, если потребуется восстановление системы, или для быстрого запуска Linux на компьютере друга, у которого установлена только Windows? В том-то и прелесть Live-дистрибутивов: вы можете найти для них множество применений.

OpenSUSE 10.2 базируется на ядре 2.6.18, *GCC 4.1.3* и *X.org 7.2* и предоставляет рабочие столы Gnome 2.16.1 и KDE 3.5.5. Что касается программ, вы найдете здесь *OOo 2.0.4* и *Firefox 2.0*. Поскольку OpenSUSE 10.2 Live работает прямо с DVD, предназначенного только для чтения, он может сохранять сделанные вами изменения или созданные вами документы только на внешних устройствах (например, USB-брелке). Помните, что если, вы, например, работаете в Live-дистрибутиве в Сети и загружаете сообщения электронной почты, они будут сохраняться в ОЗУ и утратятся при перезагрузке, поэтому перед выходом из системы убедитесь, что вы сохранили все, что скачали или создали.

Релизы OpenSUSE всегда имели довольно высокие системные требования, и версия Live – не исключение: вам потребуется как минимум 512 МБ ОЗУ. Конечно, чем больше ОЗУ, тем выше производительность, но помните, что она никогда не будет работать так же быстро, как дистрибутив, установленный на жесткий диск, потому что DVD-приводы на порядок медленнее, чем жесткие диски. Учтите, что у OpenSUSE Live нет установленного пароля администратора, поэтому если у вас потребуют ввести пароль root, просто нажмите Enter или OK.

Ознакомиться с работой в OpenSUSE вам поможет краткое руководство, представленное ниже, а если у вас возникнут проблемы при запуске или во время работы с дистрибутивом, посетите www.suseforums.net, направив туда подробности о вашем оборудовании и сообщения об ошибках. Если и здесь не повезет, посетите наши форумы на www.linuxforum.ru.



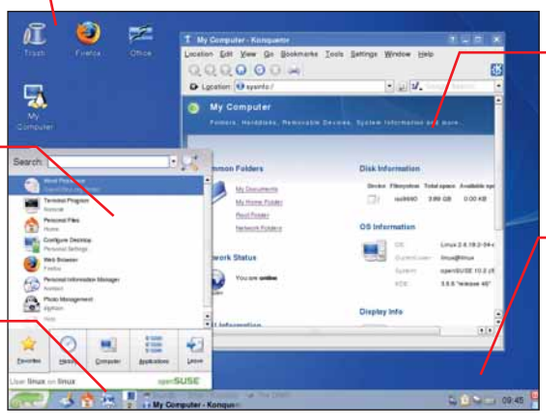
› Если вам надо перенастроить Live-сессию, поможет *Yast*.

Исследуем рабочий стол KDE в OpenSUSE

Значки
Значки на рабочем столе обеспечивают быстрый доступ к программам – здесь также можно разместить ярлыки всяческих файлов.

Главное меню
OpenSUSE снабжен модифицированным меню KDE, оно использует вкладки для разделения задач по управлению программами.

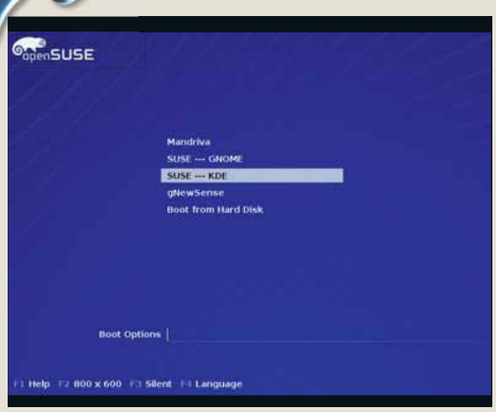
Панель
Значки для запуска программ, переключатель виртуальных рабочих столов и панель задач.



Konqueror
Файловый менеджер KDE также служит отличным быстрым web-браузером.

Системный лоток
Нажмите на красную кнопку, чтобы выйти из OpenSUSE или завершить работу.

Шаг за шагом: Работаем с OpenSUSE 10.2



1 Загрузка
Загрузите ваш ПК с DVD и выберите из загрузочного меню SUSE - Gnome или SUSE - KDE как рабочий стол по умолчанию. Если вы никогда еще не использовали Linux, начните с KDE.

2 Работа
После загрузки перед вами откроется рабочий стол. Познакомьтесь с ним, пощелкав на нем мышкой там, где захочется – поскольку дистрибутив работает с DVD, вы не причините ему никакого ущерба. А если что-то напутаете – просто перезагрузите ПК!

Дистрибутив Linux

GNewSense

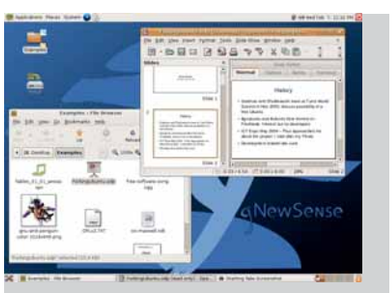
Вообразите такую картину: вы сидите себе за компьютером и работаете в Linux, используя свои любимые программы. Заходите в YouTube за самыми свежими классными новинками, но…оказывается, что их нельзя просмотреть! YouTube обновила свои Flash-приложения, и теперь они требуют новую версию проигрывателя, а выхода его под Linux ждать еще ой как долго! Опаньки, думаете вы, и решаете с горя сыграть в *BZFlag*. Бумс! Ваш X-сервер завершает работу и оставляет вас в текстовом терминале, потому что ваша лицензия на чисто двоичные видеодрайверы истекла. Тут вы, обозленный, хотите успокоиться, послушав какой-нибудь мирный файл .WMA через *MPlayer*.

Тук-тук! А это полиция под дверью, которая ловит вас на использовании Audio® Technology®, запатентованной Microsoft® исключительно для Microsoft OS. И пока вас везут в полицейский участок, у вас в голове звучит голос бородатого хакера: «А где же наша свобода?» Ричард Столлмен [Richard Stallman] напоминает вам, как легко попасться, полагаясь только на проприетарные программы: ведь их могут отнять у вас в любой момент.

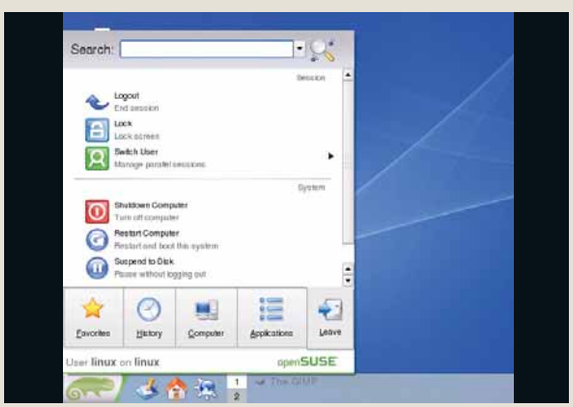
Да, эта ситуация маловероятна, но все же мы нередко забываем о важности свободы ПО. Возможно, кому-то Столлмен и действует на нервы, но благодаря его неутомимым трудам почти у каждой проприетарной программы есть свободная альтернатива. А как насчет дистрибутивов Linux? Даже бесплатные-как-пиво дистрибутивы сообществ вроде Ubuntu включают двоичные прошивки и предоставляют легкий доступ к несвободным программам. Противостоять искушению очень сложно.

GNewSense – это переделанная версия Ubuntu 6.06, из которой убрали все несвободное ПО. Она не только унаследовала превосходный дизайн Ubuntu, основанный на Debian, но и изгнала проприетарные части, не позволяющие вашей системе быть стопроцентно свободной. Устанавливая и распространяя GNewSense, вы подлинно способствуете продвижению идеи свободы ПО плюс получаете гарантию: все, что есть на вашей машине, можно выложить для открытого доступа, и с точки зрения этики все тип-топ.

Для запуска GNewSense загрузите компьютер с нашего DVD, и выберите его в загрузочном меню (потребуется 256 МБ ОЗУ). Подобно Ubuntu, это Live-дистрибутив с рабочим столом Gnome – можете поиграть с ним или щелкнуть дважды на значке Install на рабочем столе, если вы хотите использовать его постоянно (потребуется не менее 5 ГБ свободного пространства на жестком диске). Программа установки GNewSense отличается изумительной простотой использования – просто следуйте подсказкам, и у вас уйдет на все не более получаса. А уж если возникнут проблемы, загляните на www.gnewsense.org и www.ubuntuforums.org. »

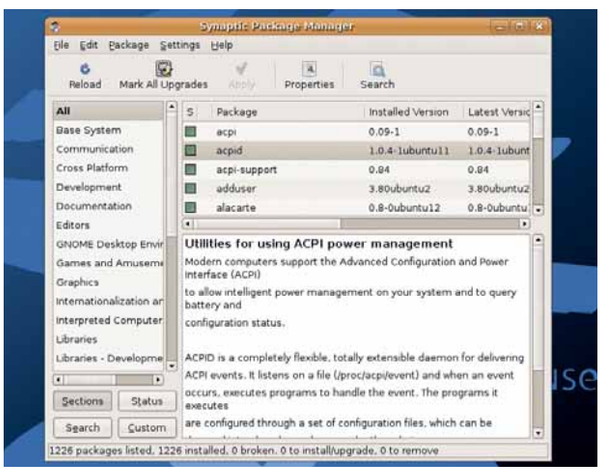


› В папке Examples на рабочем столе – информация о GNewSense.



3 Завершение

Закончив работу, щелкните по пиктограмме с зеленым гекконом, перейдите на вкладку Leave и выберите один из предложенных вариантов. В случае перезагрузки не забудьте вынуть DVD, когда компьютер перезапустится.



› Добавлять новое ПО блаженно-просто благодаря Synaptic, графической надстройке над менеджером пакетов APT от Debian.

Среда рабочего стола

Xfce 4.4

Некогда CDE-подобный модуль *FVWM*, *Xfce* вырос до превосходной среды рабочего стола на основе *GTK 2* и стал облегченной альтернативой Gnome. Обычно *Xfce* запускается в два раза быстрее, чем Gnome, и его основные компоненты (панель, файловый менеджер и т.д.) требуют значительно меньше памяти. Если на вашем компьютере меньше 256 Мб ОЗУ, и вы считаете, что Gnome и KDE тормозят, было бы мудро дать шанс *Xfce*.

Xfce 4.4 шагнул далеко вперед по сравнению с прошлым крупным релизом (4.2), применив новый шустрый менеджер файлов под названием *Thunar* взамен старичка *Xffm*.

Добавились две новых утилиты *Mousepad* и *Orage* – соответственно, текстовый редактор и календарь. Разработчикам также удалось увеличить стабильность расширений панели: если один из апплетов «упадет», он не потащит за собой всю панель целиком. А если вы – поклонник значков на рабочем столе, радуйтесь: 4.4 включил и эту желанную функцию.

Чтобы установить *Xfce*, сначала удалите все ранее установленные версии через менеджер пакетов, затем скопируйте файл **xfce4-4.4.0-installer.run** из раздела **Desktop/Xfce** нашего DVD в свою домашнюю директорию. Откройте терминал и введите

```
sh xfce4-4.4.0-installer.run
```

Далее следуйте подсказкам, и *Xfce* будет скомпилирован и установлен (если вы плохо представляете, куда его запихнуть, оставьте путь по умолчанию: **/home/имяпользователя/local**). О нехватке какой-нибудь зависимости вам сообщат – подберите ее с помощью менеджера пакетов. Покончив с этим, выйдите из среды рабочего стола, чтобы вернуться в окно входа в систему, выберите сессию failsafe или terminal и, чтобы загрузиться в *Xfce*, введите команду:

```
~/local/bin/startxfce4
```

› Команда *Xfce* заменила старенький файловый менеджер шикарным *Thunar*.

Документация

Linuxpedia, Ответы и PDF-файлы

На DVD этого месяца не просто набор программ, а еще и солидная документация по всем вопросам, связанным с Linux. Здесь вы найдете достаточно материала, расширяющего ваши знания и достойного сохранения в качестве справочного руководства. Если вы прочтете все включенное в наш диск, гарантируем: ваш мозг станет на 24% Linux'овее.

В первую очередь, у нас имеется Linuxpedia, колоссальный сборник фактов в формате HTML. В процессе чтения этого журнала вы, верно, сталкивались с такими терминами и техноязом, которые ставили вас в тупик. Когда мы говорим об инструментах командной строки, можно, по крайней мере, скоренько ввести в терминал **man** и понять, что к чему, но как быть с дистрибутивами, программами или личностями в Linux?

В разделе **Help/Linuxpedia** нашего DVD вы найдете более 500 страниц информации по Linux в формате HTML*, легко просматриваемых из оболочки диска (для этого откройте **index.html** в корневой директории DVD). Мы взяли лучшую часть информации по Linux из Wikipedia и переформатировали ее в более легкий формат HTML, его можно запросто скинуть на PDA и изучать Linux на ходу.

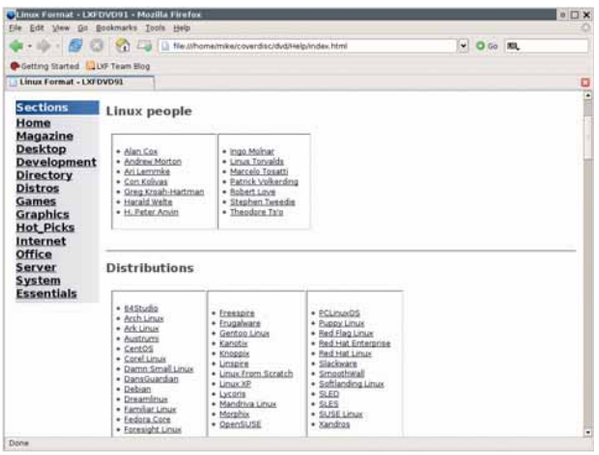
Если только вы не провели последние десять лет на Рокколле [Rockall – необитаемый скалистый остров в Северной Атлантике, – *прим. пер.*], то вы наверняка знаете все о Wikipedia, но на всякий случай сообщаем: Wikipedia (**www.wikipedia.org**) – это создаваемая совместными усилиями онлайн-энциклопедия, в которую пользователи всего мира вносят посильный вклад, создавая и редактируя статьи. Поскольку это может сделать каждый, совершенно свободно, самая лучшая информация выбирается наверх по принципу «выживает лучший». Если кто-то добавит неверную информацию, более знающие пользователи ее исправят, и останется только лучшее.

Как следствие, в Wikipedia полным-полно качественной информации по Linux. На 500 HTML-страницах с DVD вы найдете россыпи объяснений по всевозможным темам, включая дистрибутивы Linux, технологии ядра, ПО и важнейших игроков в мире Open Source. Когда в следующий раз увязнете, недоумевая, кто такой Алан Кокс [Alan Cox] или зачем нужен *Klibc*, ищите ответы здесь. И, естественно, так как содержание Wikipedia свободно распространяется по лицензии Свободной Документации GNU, вы можете передать ее другим пользователям Linux, если им потребуется помощь!

Ответы и PDF-файлы

Второе блюдо нашего мега-пиршества документации – Ответы, они тоже находятся в разделе **Help**. Ежемесячно в Башни **LXF** приходят вопросы от пользователей, у которых возникли проблемы с Linux, и мы стараемся решить все эти проблемы в разделе Ответы. Мы взяли разделы Ответов из шести последних выпусков *Linux Format* (выпуски с 85 по 90) и включили весь этот текст* на наш DVD – на одной чудесной большой HTML-странице, чтобы вы смогли найти решение за считанные секунды.

Наш раздел Ответы охватывает множество тем, связанных Linux, и если в повседневном использовании этой ОС вы столкнулись с неким затруднением, стоит глянуть сюда.



› **Ваш портативный источник знаний о Linux, отточенных доброхотами в Wikipedia.**

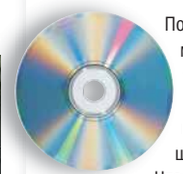
Сейчас на странице 60 вопросов и ответов, мы будем обновлять их ежемесячно, так что следите за этим разделом!

И, наконец, в разделе **Журнал/PDF** нашего DVD вы найдете три статьи из предыдущих номеров *Linux Format*, включая обзор проекта OpenSolaris (LXF79). Сможет ли флагманская ОС Sun с открытым кодом помочь им в конкуренции с Linux? Или это просто попытка погарцевать на гребне волны Open Source? Статья из LXF81 рассматривает возможности Linux в образовании, а в последнем PDF показано, как воскресить классические игры, используя эмуляторы Spectrum, C64, SNES и прочей золотой старины.

*** – на английском языке**

› **Скучаете по доброму старому Пентагону? Прочтите об эмуляторах в статье из LXF69.**

И наконец...



После всех этих дистрибутивов и документации, дайте мозгу заслуженный отдых – тут вам помогут блистательные игры для Linux. Главное развлечение этого месяца – *Ultimate Stunts*, забавные 3D-гонки, подвергающие испытаниям ваш талант водителя на петляющих и карабкающихся наверх извилистых трассах. Чтобы установить игру, скопируйте файл **ultimatestunts-srcdata-0.7.0.1.tar.gz** из раздела **Games/Ultimate_Stunts** на жесткий диск, распакуйте, и в полученной директории введите (от имени суперпользователя)

```
./configure
make && make install
```

Если все сделано правильно (требуется *SDL* и *OpenAL*), просто введите **ustunts**, и игра запустится. Хотя это пока еще не релиз 1.0, играть в нее все равно здорово, и она должна помочь вам выпустить пар, накопленный в пробках по дороге домой с работы. LXF

› **If (колес_оторванных_от_земли > 3) { ваша_крутизна++; }**

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК !

Информация о диске

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Содержание LINUX FORMAT DVD Апрель 2007

Сторона 1

Аудио
AlsaPlayer – аудиопроигрыватель
Aqualung – аудиоплеер для Linux/FreeBSD
Sweep – аудиоредактор с поддержкой LADSPA

Рабочий стол
AbiWord – свободный текстовый процессор
AmiWM – оконный менеджер, эмулирующий вид Amiga Workbench
F-Spot – фотоальбом
gDesklets – программа для отображения десклетов
MLVWM – оконный менеджер с видом MacOS Classic
PeaZip – кроссплатформенный архиватор
ROX – легкий и быстрый рабочий стол
Xfce – рабочее окружение для Unix

Разработка
Anjuta – современная IDE для C/C++
Kdevelop – среда разработки для KDE
Kscope – инструмент для изучения и правки кода на C
Mono – реализация .NET
MonoDevelop – среда разработки для Mono

Дистрибутивы
gNewSense – версия Ubuntu, содержащая только свободные компоненты
Mandriva – LiveCD с предварительной версией рабочего стола Metisse
OpenSUSE Live – LiveCD на основе OpenSUSE

Игры
Brutal_Chess – трехмерные шахматы
Diameter – космическая аркада
DuckMaze – лабиринт против уточки
Phantomas – ремейк одноименной игры 1986 года
Ultimate_Stunts – симулятор автокаскадера

Графика
Inkscape – векторный редактор
K-3D – 3D-редактор
Tripod – утилита для управления фотоальбомами в iPod

Справка
Решенные проблемы из рубрики «Ответы»
Linuxpedia – 500 страниц о Linux из Википедии

Интернет
Ktorrent – BitTorrent-клиент для KDE
Netrik – консольный веб-браузер
SeaMonkey – комплект программ для работы в Интернете

Безопасность
ClamAV – свободный антивирус
Firestarter – графический интерфейс настройки брандмауэра
John_the_Ripper – утилита для тестирования паролей

Сервер
eyeOS – Web-основанная рабочая среда
FlameRobin – утилита администрирования Firebird DBMS
phpMyBackupPro – инструмент резервного копирования баз MySQL

Система
Metisse – оконный менеджер с трехмерными эффектами
QEMU – свободный эмулятор процессора
Recoll – поисковый инструмент
SMBNetFS - «сетевое окружение» для Linux/FreeBSD

Поставляется вместе с журналом LINUXFORMAT номер 4(91) Апрель 2007

Сторона 2

Sabayon Linux 3.3

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен на Уральском электронном заводе., 620066, Россия, г. Екатеринбург, ул. Комвузовская 17-203, Лицензия МПТР России ВАФ № 77-15

DVD

СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Содержание DVD

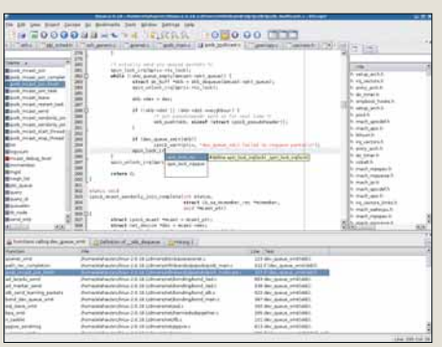
ЖУРНАЛ
Blender .....Файлы из учебника Blender
Gtk .....Код примеров статьи.
JavaEE .....Код примеров статьи.
Mono.....Код примеров статьи.
PDFs .....Статьи из предыдущих LXF.
Roundup.....Системы управления версиями.
SugarCRM.....Код примеров статьи.
Unix API .....Код примеров статьи.

ЗВУК
AlsaPlayer .....Медиа-плейер.
Aqualung.....Музыкальный плейер.
Sweep.....Звуковой редактор.

РАБОЧИЙ СТОЛ
AbiWord.....Текстовый редактор.
AmiWM .....Оконный менеджер а-ля Workbench.
F-Spot .....Менеджер фотографий.
GDesklets.....Мини-утилиты рабочего стола.
MLVWM .....Классический оконный менеджер в стиле Mac.
PeaZip .....Архиватор файлов.
ROX.....Рабочий стол в стиле RISC OS.
Skinning.....Скины Gnome и KDE.
Xfce .....Среда рабочего стола.

РАЗРАБОТКА
Anjuta .....IDE на основе GTK 2.
KDevelop .....Интегрированная среда разработки.
KScope .....Редактор исходного кода.
Mono.....Открытая реализация Microsoft .NET.
MonoDevelop.....C# IDE.

ДИСТРИБУТИВЫ
gNewSense.....Устанавливаемый и Live-дистрибутив на базе Ubuntu.
Mandriva.....Live-дистрибутив.
OpenSUSE.....Live-дистрибутив, спонсируемый Novell.



› *KScope*: IDE, разработанная для крупных проектов на языке C – например, ядра Linux.

ИГРЫ
Brutal Chess .....Симпатичная головоломка.
Diameter.....Аркада.
DuckMaze.....Утка в лабиринте.
Phantomas.....Адвентура.
Ultimate Stunts.....Гонки.

ГРАФИКА
Inkscape .....Редактор векторной графики.
K-3D.....Система 3D моделирования и анимации.
Tripod .....Менеджер фотографий iPod.

› Шахматные фигуры в *Brutal Chess* выглядят до того реально, что немудрено по ошибке сунуть руку в экран.

HELP
Answers.....Решение проблем в Linux.
Linuxpedia .....Энциклопедия Linux.
RUTE.....Руководство по администрированию Linux.

HOTPICKS
Audacity.....Звуковой редактор.
Breve .....Инструмент визуализации AI.
Bugle.....Отладчик OpenGL.
Dagii .....Интерпретатор приключенческих игр.
ExactImage .....Библиотека обработки изображений.
JSHoldem .....Web-игра.
Klavaro .....Тренажер клавиатуры.
KX Generator .....Инструмент настройки X-сервера.
Qemulator .....Интерфейс пользователя Qemu.
SystemRescueCD ....Инструмент восстановления Linux-системы.

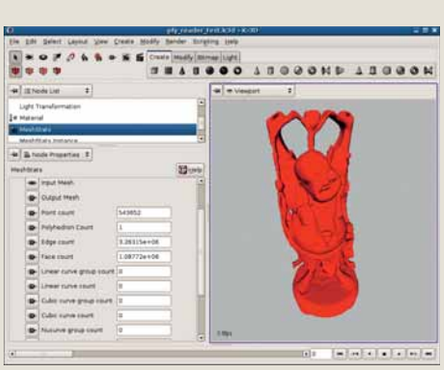
ИНТЕРНЕТ
KTorrent.....Клиент BitTorrent.
Netrik .....Текстовый браузер.
SeaMonkey.....Пакет для работы в Интернете.

БЕЗОПАСНОСТЬ
ClamAV .....Антивирус.
Firestarter.....Визуальный брандмауэр.
John the Ripper .....Взломщик паролей.

СЕРВЕР
EyeOS .....Web-среда рабочего стола.
FlameRobin.....Интерфейс для баз данных.
PhpMyBackupPro ....Web-приложение для создания резервных копий баз MySQL.
SugarCRM.....CRM-система.

СИСТЕМА
Metisse.....Система эффектов рабочего стола.
Qemu .....Эмулятор ПК.
Recoll.....Утилита поиска по тексту.
SMBNetFS.....Браузер для сетей Windows.

ГЛАВНОЕ
ATI driver.....Графический драйвер.
Avifile.....Библиотека чтения/записи AVI файлов.
Bash.....Командный интерпретатор.
CheckInstall .....Программа для создания двоичных пакетов.
Coreutils .....Утилиты командной строки.
CSV.....Содержание дисков LXF.
GLib .....Низкоуровневая библиотека.
Glibc.....Библиотека GNU C.
GTK.....GUI-инструментарий.
HardInfo .....Информация и оценка системы.
Jigdo.....Создатель ISO-образов.
Kernel .....Свежий релиз ядра Linux.
LibXML2 .....Анализатор и набор инструментов XML.
Ncurses.....Оконный инструментарий текстового режима.
Nvidia driver .....Графический драйвер.
Python.....Язык программирования.
Rawrite .....Программа записи изображений на дискеты.
SBM .....Smart Boot Manager.
SDL .....Библиотека мультимедиа.



› Создайте довольного Будду и многое другое в K-3D – первоклассной системе 3D-моделирования.

Линуксцентр представляет авторизованные курсы Mandriva Linux.

Курсы не привязаны к конкретному дистрибутиву и сертифицированы международной организацией LPI - Linux Professional Institute (www.lpi.org).

★ LNX70: Введение в Linux.

Предназначен для пользователей, которые хотят приобрести навыки использования Linux. В рамках курса рассматриваются вопросы миграции на Linux и даются базовые знания для пользователей.

Цель курса: Обучить пользователя оптимальной работе с Linux.
Требования к соискателям: Базовая компьютерная подготовка.

★ LPI101: Администрирование Linux для начинающих.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования Linux. Рассматриваются следующие вопросы: конфигурирование системы, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

Цель курса: Подготовить системного администратора Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI101.

Требования к соискателям: Знания по Linux в объеме курса LNX70.

★ LPI102: Сетевое администрирование Linux.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования сетей на базе Linux. Рассматриваются следующие вопросы: конфигурирование периферийных устройств, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

Цель курса: Подготовить системного администратора к управлению сетями на базе Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI102.

Требования к соискателям: Знания по Linux в объеме курса LPI101.

★ LPI201: Администрирование Linux для специалистов.

Курс предназначен для системных администраторов, которые хотят повысить свою квалификацию в области управления сложными Linux-системами. Рассматриваются следующие вопросы: настройка ядра Linux, восстановление системы, восстановление файловой системы, настройка RAID, сопровождение системы, автоматизация задач администрирования.

Цель курса: Подготовить системного администратора к управлению сложными Linux-системами в соответствии с требованиями к соискателям на сдачу экзамена LPI201.

Требования к соискателям: Знания по Linux в объеме курса LPI101 или LPI102.

★ LPI202: Сетевое администрирование Linux для специалистов.

Курс предназначен для опытных системных администраторов, которые хотят усовершенствовать навыки управлениями сетевыми сервисами на базе Linux. Рассматриваются следующие вопросы: углубленное конфигурирование сети, настройка служб электронной почты и новостей, расширенная настройка DNS, обслуживание интернет-сервера, настройка DHCP, NIS, LDAP и PAM, управление безопасностью сетевых служб, контроль за трафиком.

Цель курса: Подготовить системного администратора к управлению сложными сетевыми сервисами в соответствии с требованиями к соискателям на сдачу экзамена LPI202.

Требования к соискателям: Знания по Linux в объеме курса LPI101.

Узнайте подробности и закажите курсы на сайтах - www.linuxcenter.ru и www.unixedu.ru

Mandriva

Linux center
www.linuxcenter.ru

Mandriva Linux 2007
Дистрибутив Mandriva (ранее, до слияния с Connectiva называвшийся Mandrake), по праву считается идеальным для неподготовленного пользователя. Mandriva Linux невероятно прост в установке благодаря современному графическому установщику. Гордость дистрибутива – комплект графических утилит для настройки системы, объединенные в Mandriva Control Center. Установив Mandriva, вы сможете настроить абсолютно все, от звуковой карты до файерволла, не прибегая к помощи командной строки.

Cedega
Продукты Mandriva Linux 2007 Discovery и PowerPack включают TransGaming Cedega, позволяя запускать Windows-версии игр.

TransGaming Technologies создала уникальную технологию, позволяющую разработчикам собирать игры для одной системы и затем легко запускать их на других.
Технология TransGaming уникальна тем, что разработчикам не нужно думать о поддержке кроссплатформенности. TransGaming Cedega позволяет запускать популярнейшие игры, такие как Battlefield 2, Elder Scrolls IV: Oblivion, World of WarCraft, Civilization IV, Madden 2006 и множество других. Нет лучшего способа играть в Linux!

LinDVD
Продукты Mandriva Linux 2007 Discovery и PowerPack включают InterVideo LinDVD, позволяя просматривать DVD-диски абсолютно легально и без долгой настройки.
InterVideo -- лидирующий производитель программного обеспечения для просмотра DVD, а также мультимедиа-приложений.

Ia Ora
Последний раз Мандрива меняла тему рабочего стола ещё в версии 9.1, это была Galaxy. Для 2007 мы разработали с нуля новую тему назвав её Ia Ora ("здравствуйте" на языке франкоговорящих полинезийцев), поменяв там всё начиная от рамок окна заканчивая виджетами и иконками. Новая тема проще и легче, более яркая, хоть и используется всего несколько цветов.

3D
Поддержка нового трехмерного рабочего стола (AIGLX или Xgl) и новая утилита для его настройки (drak3d): Mandriva – единственный дистрибутив, предоставляющий легкий доступ к обеим технологиям, так что совместимость с оборудованием настолько широка, насколько это вообще возможно. Drak3d может сам определить лучший 3D-сервер. AIGLX и Xgl доступны в KDE и Gnome.

Новый RPMdrake
Mandriva разработала лучшую утилиту для установки, удаления и обновления программ. Новый RPMdrake дает вам информацию о том, какие пакеты у вас установлены, какие могут быть установлены, что рекомендует Mandriva, какие обновления доступны и т.д.

Cedega

Invictus Firewall
Mandriva разработала Invictus Firewall, высоконадежный брандмауэр для кластерных систем. Множество брандмауэров на машинах может быть настроено за считанные минуты, без отрыва их пользователей от работы.

CRM, ERP
Tiny ERP является бесплатным программным продуктом для интеграции данных и повышения продуктивности. Включен в Powerpack+. Пакет обрабатывает бизнес-процессы, продажи, финансы, поддерживает цепочки и менеджмент проекта... TinyERP дает возможность SOHO-пользователям наблюдать за их бизнесом.

Invictus Firewall

Joomla – открытая система управления контентом (CMS). Используется для всего, начиная с простых веб-сайтов и заканчивая сложными корпоративными приложениями. Мы выбрали Joomla потому, что она проста в установке, настройке и при этом надежна.

BitDefender
BitDefender предоставляет решения для защиты сетей (антивирус, анти-спам, анти-spyware) для корпоративных и серверных нужд. BitDefender отслеживает трафик, блокируя вирусы, троянских коней и другие вредоносные приложения, защищая пользователей и их данные.

bitdefender
secure your every bit

BitDefender – это:

Комплексная технология анти-спама
Управление пользователями и группами
Совместимость с SNMP
Продвинутая система обновления

Закажите продукты Mandriva (Mandrakelinux) в Линуксцентре или у наших партнеров !
www.linuxcenter.ru/mandriva

Линуксцентр представляет авторизованные курсы Mandriva Linux, сертифицированные LPI.org
www.linuxcenter.ru/training

LINUX FORMAT
Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Елена Толстякова, Иван Мищенко
Переводчики
Александр Бикмеев, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Валентин Развозжаев, Сергей Супрунов, Александр Черных, Юлия Шабунио
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____
Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Литературный редактор
Ребекка Смолли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Эндрю Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

Авторы
Марк Бейн (Mark Bain), Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Д-р Крис Браун (Dr Chris Brown), Энди Ченнел (Andy Channelle), Ричард Драммонд (Richard Drummond), Йон Филлипс (Jon Phillips), Рейчел Проберт (Rachel Probert), Натан Сандерс (Nathan Sanders), Евгений Балдин, Андрей Боровский, Андрей Прахов, Александр Супрунов, Сергей Супрунов, Алексей Федорчук, Антон Черноусов, Денис Шмаков, Илья Шпаньков
Художественные ассистенты: Зигги Бейкер (Ziggi Baker), Эми Кустон (Amie Causton)
Фотографии: Роб Скотт (Rob Scott)
Иллюстрации: Нейл Бартлет (Neil Bartlett), Пол Блехфорд (Paul Blachford), Крис Винн (Chris Winn)

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Представительство в Москве:
пр.Мира, 161, телефон +7(495) 799-18-63, +7(495)136-88-45
Email: info@linuxformat.ru, Web: www.linuxformat.ru

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензией Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

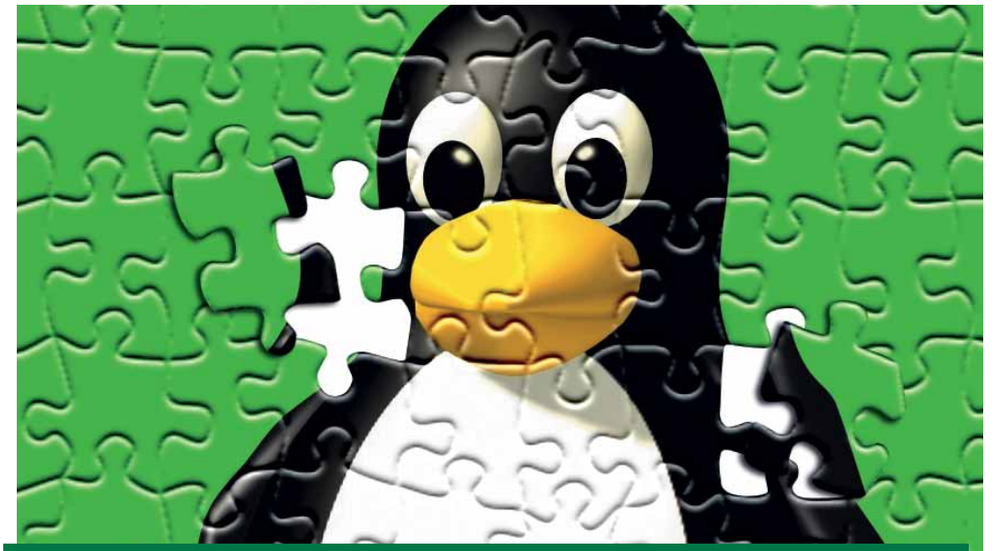
За содержание рекламных материалов редакция ответственности не несет.

Linux-зарегистрированная торговая марка Линуса Торвальдса (Linus Torvalds). Название «GNU/Linux» заменяется на «Linux» в целях сокращения. Остальные торговые марки являются собственностью их законных владельцев.
Linux Format является торговой маркой Future Publishing Ltd (Future plc group company).

За информацией о журналах, издаваемых Future plc group company, обращайтесь
http://www.futureplc.com

© Linux Format 2005
© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO •SAN FRANCISCO



в мае

АПГРЕЙД по всем правилам

Четырехядерный процессор, 4 ГБ памяти и 1 500 ГБ дискового пространства: узнайте, как собрать убойный компьютер, 100%-но совместимый с Linux!

Wine: потерянное руководство

Ветераны раскроют фокусы, с помощью которых они заставляют работать в Wine приложения Windows и игры.

SpamAssassin: доступно и понятно

Устали от постоянных предложений увеличить или заработать? Заблокируйте их, как истинный профи.

Ян Мердрок

Побеседуем с основателем проекта Debian

Содержание последующих выпусков может меняться без уведомления

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА В ЛИНУКСЦЕНТРЕ

Сколько стоит подписка?

Подписка на журнал «Linux Format» 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь) стоит 1800 рублей
Подписка на журнал «Linux Format» 6 номеров (июль, август, сентябрь, октябрь, ноябрь, декабрь 2006 года) стоит 900 рублей

Как оформить подписку?

Чтобы оформить подписку на журнал «Linux Format», необходимо зарегистрироваться в интернет-магазине Linuxcenter.Ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал «Linux Format» 12 номеров 2006 года», или товар «Подписка на журнал «Linux Format» второе полугодие 2006 года», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Как оплатить подписку?

– по выставленному счету (для юридических лиц)
– по квитанции в любом отделении Сбербанка

Плюсы подписки

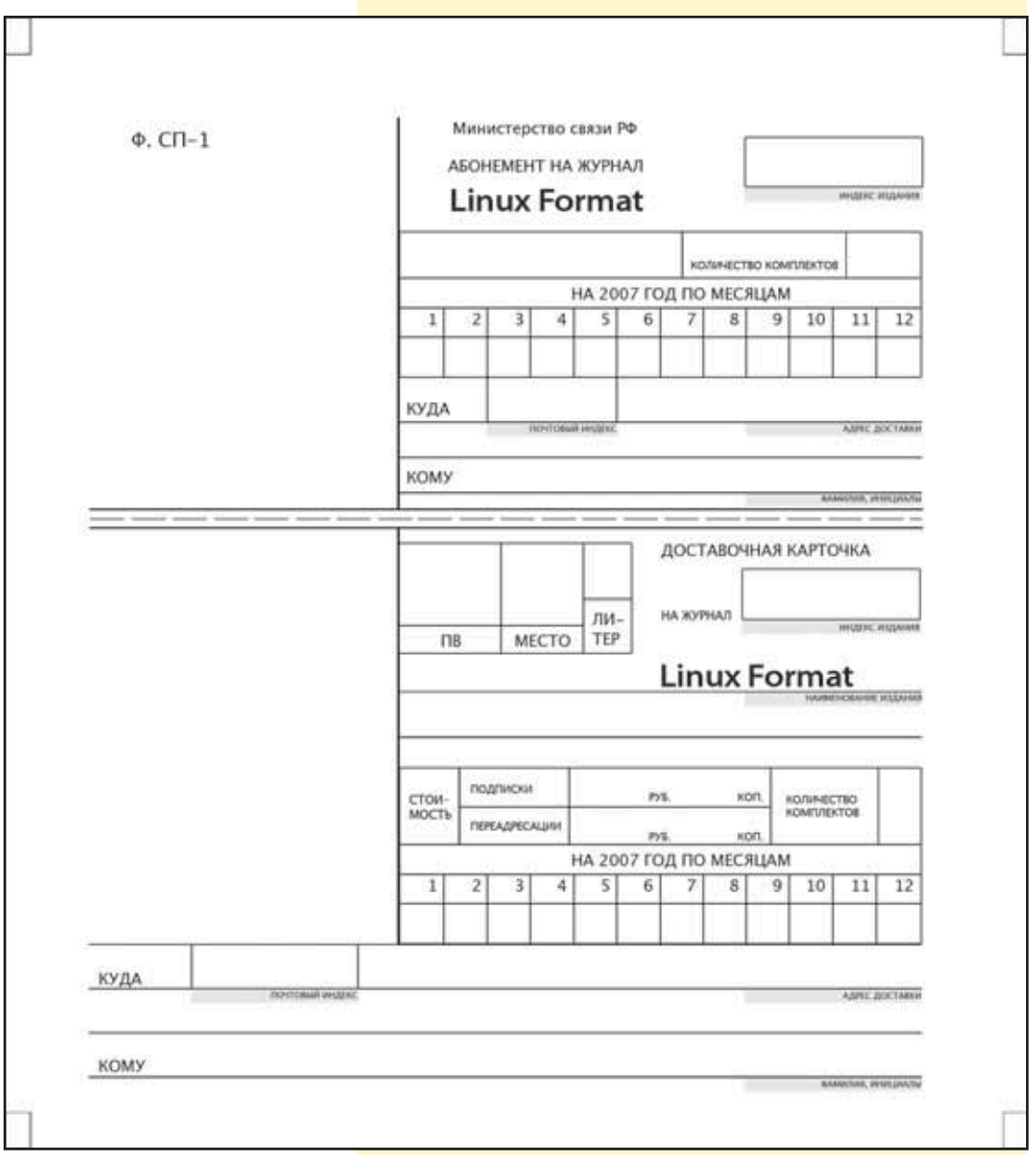
- подписка дешевле!
- гарантированное получение нового номера журнала!

ПОДПИСКА - 2007!

ПОДПИСКА ПО КАТАЛОГАМ РФ

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882

Каталог «ПРЕССА РОССИИ» – подписной индекс 87974



Ф. СП-1

Министерство связи РФ

АБОНЕМЕНТ НА ЖУРНАЛ

Linux Format

индекс издания

количество комплектов

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА

почтовый индекс

адрес доставки

КОМУ

фамилия, инициалы

ДОСТАВОЧНАЯ КАРТОЧКА

НА ЖУРНАЛ

индекс издания

ПВ	МЕСТО	ЛИТЕР

Linux Format

наименование издания

СТОИМОСТЬ	ПОДПИСКИ	РУБ.	КОП.	КОЛИЧЕСТВО КОМПЛЕКТОВ
	ПЕРЕАДРЕСАЦИИ	РУБ.	КОП.	

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА

почтовый индекс

адрес доставки

КОМУ

фамилия, инициалы

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА ПО КАТАЛОГАМ СНГ И БЛИЖНЕГО ЗАРУБЕЖЬЯ

Каталог «Российская Пресса» – совместный проект Государственного предприятия «Казпочта» , Агентства «Книга-Сервис» и АРЗИ.

Блок изданий АРЗИ в национальных Каталогах Украины и Беларуси. В Азербайджане, Армении, Грузии, Киргизии, Узбекистане и Молдове - по изданиям, включенным в Объединенный каталог, распространяемые через АРЗИ.

Азербайджан
- по Объединенному каталогу российских изданий через Предприятие по распространению печати «Гасид»(370102, г. Баку, ул. Джавадхана, 21);

Армения
- по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г.Ереван, пл.Сасунци Давида, д.2) и ЗАО «Контакт-Мамул» (375002, Г.Ереван, ул.Сарьяна, 22);

Белоруссия
- по Каталогу изданий стран СНГ через РГО «Белпочта» (220050, г.Минск, пр-т Ф.Скорины, 10);

Грузия
- по списку номенклатуры «АРЗИ»» через АО «Сакпресса» (380019, г.Тбилиси, ул.Хошараульская, 29) и АО «Мацне» (380060, г.Тбилиси, пр-т Гамсахурдия, 42);

Казахстан
- по Каталогу «Российская Пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»;

Молдавия
- по каталогу через ГП «Пошта Молдавей» (МД-2012, г.Кишинев, бул.Штефан чел Маре, 134);
- по списку через ГУП «Почта Приднестровья» (MD-3300, г.Тирасполь, ул.Ленина, 17);
- по прайс-листу через ООО Агентство «Editil Periodice» (2012, г.Кишинев, бул. Штефан чел Маре, 134).

Узбекистан
- по Каталогу «Davriy nashrlar» российские издания через Агентство по распространению печати «Davriy nashrlar» (7000029, Ташкент, пл.Мустакиллик, 5/3, офис 33);

Украина
- Киевский главпочтамт.
- Подписное агентство «KSS» Телефон/факс (044)270-62-20, 270-62-22

ПОДПИСКА НА LINUX FORMAT

Агентство "Centerpress"

Сколько стоит подписка?
Подписка на журнал "Linux Format" 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь 2007 года) стоит 1800 рублей.

Как оформить подписку?
Чтобы оформить подписку на журнал "Linux Format", необходимо зарегистрироваться в интернет-агентстве Centerpress.ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар "Подписка на журнал "Linux Format" на 2007 год 12 номеров (01-12 / 2007)", получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Агентство "Centerpress" : www.centerpress.ru

Все Плюсы подписки!
- Подписка дешевле!
- Гарантированное получение журнала!

По каталогам РФ

Каталог агентства "РОСПЕЧАТЬ" - подписной индекс

20882

Каталог "ПРЕССА РОССИИ" - подписной индекс

87974

АЛЬТЕРНАТИВНЫЕ АГЕНСТВА РФ

Агентство «Интер-Почта»
(095) 500-00-60, курьерская доставка по Москве.

Агентство «Вся Пресса»
(095) 787-34-47

Агентство «УралПресс»

- **Екатеринбург, Березовский, В. Пышма, Первоуральск тел. (343) 375-80-71, 375-84-93, 375-84-39, факс 375-62-74, info@ural-press.ru**
- **Нижний Тагил тел. (3435) 411448, 417709, ntagil@ural-press.ru**
- **Челябинск тел. (351) 262-90-03, 262-90-05, pochta@chel.surnet.ru**
- **Пермь тел. (3422) 60-24-40, 60-22-95, 60-35-42, parma-press@permonline.ru**